

**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**

**Периферійний пристрій для систем автоматизованого проектування**

Спеціальність 123 Комп'ютерна інженерія

123 – КБР.01 – 405.22010504

**Студент**

  
\_\_\_\_\_ О. М. Горішна  
*підпис*

«\_\_» \_\_\_\_\_ 202\_\_ р.

**Керівник ст. викладач**

\_\_\_\_\_ І. С. Бурлаченко  
*підпис*

«\_\_» \_\_\_\_\_ 202\_\_ 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

Горішній Ользі Михайлівні

*(прізвище, ім'я, по батькові студента)*

1. Тема кваліфікаційної роботи

Периферійний пристрій для систем автоматизованого проєктування

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

Очікуваним результатом роботи є: периферійний пристрій для систем автоматизованого проєктування. Вхідними даними роботи є: специфікація вимог, що описує характеристики зазначеного апаратного та програмного забезпечення.

4. Перелік питань, що підлягають розробці

1) Аналітичний огляд сучасних технологій периферійних пристроїв для систем автоматизованого проєктування;

2) Огляд і аналіз існуючих аналогів та публікацій в галузі периферійних пристроїв для систем автоматизованого проєктування;

3) Визначення технічних вимог та постановка задачі для розробки периферійного пристрою, враховуючи специфіку систем автоматизованого проєктування;

4) Моделювання та технічне проєктування периферійного пристрою;

5) Програмна реалізація периферійного пристрою.

5. Перелік графічних матеріалів

- Блок-схема алгоритму роботи пристрою;
- Схема підключення пристрою.

6. Завдання до спеціальної частини

Проаналізувати вплив дотримання стандартів безпеки та гігієни праці на продуктивність та здоров'я працівників у системах автоматизованого проектування. Зокрема, розглянути використання периферійних пристроїв та їхній внесок у зменшення ризиків травм та втоми, базуючись на нормативних документах.

7. Консультанти:

Консультант	Кафедра (організація)	Частина роботи
Макарова О. В. ст. викл.	кафедра екології Медичного інституту ЧНУ імені Петра Могили	Спеціальна частина з охорони праці


Керівник роботи

ст. викладач кафедри комп'ютерної інженерії Бурлаченко Іван Сергійович  
(посада, прізвище, ім'я, по батькові)

\_\_\_\_\_  
(підпис)

Завдання прийнято до виконання

Горішна Ольга Михайлівна  
(прізвище, ім'я, по батькові студента)

  
\_\_\_\_\_  
(підпис)


Дата видачі завдання « \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

**КАЛЕНДАРНИЙ ПЛАН**  
**виконання кваліфікаційної роботи**

Тема: Периферійний пристрій для систем автоматизованого проєктування

№	Найменування роботи	Початок	Закінчення	Примітки
1	Розробка та затвердження завдання на виконання КР	11.12.2023	12.12.2023	Виконано
2	Огляд літератури за темою роботи	15.12.2023	15.01.2024	Виконано
3	Складання календарного плану БКР	16.01.2024	01.02.2024	Виконано
4	Аналіз предметної області	01.02.2024	01.03.2024	Виконано
5	Розробка проєктних рішень	01.03.2024	20.03.2024	Виконано
6	Моделювання та конструювання АПЗ	20.03.2024	20.04.2024	Виконано
7	Перевірка працездатності, тестування та апробація розробленого АПЗ, аналіз результатів тестування	20.04.2024	01.05.2024	Виконано
8	Відгук керівника КР	06.06.2024	15.06.2024	Виконано
9	Оформлення БКР та презентації	01.05.2024	28.05.2024	Виконано
10	Перший попередній захист	28.05.2024	28.05.2024	Виконано
11	Другий попередній захист	05.06.2024	05.06.2024	Виконано
12	Рецензування	06.06.2024	15.06.2024	Виконано
13	Завершення оформлення КР та презентації	06.06.2024	15.06.2024	Виконано
14	Захист бакалаврської кваліфікаційної роботи	24.06.2024	26.06.2024	Виконано

Розробив здобувач ВО Горішна Ольга Михайлівна  
(прізвище, ім'я, по батькові)

  
(підпис)

« \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

Керівник роботи ст.викл. Бурлаченко Іван Сергійович  
(посада, прізвище, ім'я, по батькові)

\_\_\_\_\_  
(підпис)

« \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

## АНОТАЦІЯ

до кваліфікаційної бакалаврської роботи  
«Периферійний пристрій для систем автоматизованого проектування»  
Студент 405 гр.: Горішна Ольга Михайлівна  
Керівник: ст. викладач Бурлаченко Іван Сергійович

Ця робота присвячена розробці периферійного пристрою для систем автоматизованого проектування (САПР), який складається з приймача сигналу та передавача сигналу. Головною метою проекту є створення пристрою, що дозволяє інтегрувати управління САПР за допомогою бездротової технології та забезпечити зручність користувача.

Система включає два Bluetooth модулі HC-05, Arduino Pro Micro для приймача та Arduino Micro для передавача. В пристрої використовуються дві кнопки для емуляції лівого та правого кліку мишки, а також сенсор MPU6050 для відслідковування руху.

Приймач сигналу на базі Arduino Pro Micro з'єднаний з Bluetooth модулем HC-05, що приймає сигнали від передавача. Передавач на базі Arduino Micro також оснащений Bluetooth модулем HC-05, який відправляє сигнали на приймач. MPU6050 сенсор відстежує рухи пристрою та перетворює їх у команди для управління курсором мишки. Дві кнопки використовуються для емуляції кліків мишки, забезпечуючи інтерактивність користувача з системою САПР.

Програмний код написано мовою програмування Arduino (C++). Він обробляє сигнали від сенсора MPU6050, передає дані через модулі Bluetooth і виконує команди від кнопок.

В результаті роботи створено функціональний прототип периферійного пристрою для САПР, який покращує взаємодію користувача з системою автоматизованого проектування, забезпечуючи бездротове управління та підвищуючи загальну зручність використання.

**Ключові слова:** *Arduino, C++, MPU6050, периферійний пристрій, Bluetooth, синдром зап'ястного каналу*

## **ABSTRACT**

of the Bachelor's Thesis

" Peripheral Device for Computer-Aided Design Systems"

Student: Olha Horishna

Supervisor: Senior lecturer Ivan Burlachenko

This work is dedicated to the development of a peripheral device for computer-aided design (CAD) systems. The main goal of the project is to create a wireless device that enhances interactivity and ease of use with CAD systems.

The developed system consists of a signal receiver and a signal transmitter. The receiver uses an Arduino Pro Micro microcontroller and an HC-05 Bluetooth module, while the transmitter uses an Arduino Micro with an HC-05 Bluetooth module. Two buttons are provided for left and right mouse clicks, and an MPU6050 sensor tracks the device's movements.

The program code is written in the Arduino programming language (C++). It processes signals from the MPU6050 sensor, transmits data via Bluetooth modules, and executes commands from buttons.

As a result of the work, a functional prototype of a CAD peripheral was created, which improves user interaction with the CAD system by providing wireless control and increasing overall usability.

**Keywords:** *Arduino, C++, MPU6050, peripheral device, Bluetooth, carpal tunnel syndrome*

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ .....	4
ВСТУП .....	5
1 АНАЛІТИЧНИЙ ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ ТА СКЛАДОВИХ СИСТЕМИ .....	7
1.1 Периферійні пристрої в САД .....	7
1.2 Наявні альтернативні рішення.....	8
1.3 Технологічне рішення .....	9
1.4 Вибір комплектуючих .....	11
1.5 Вимоги до апаратно- програмного комплексу .....	19
1.6 Висновки до розділу 1 .....	20
2 РОЗРОБКА АПАРАТНОЇ ЧАСТИНИ.....	22
2.1 Апаратне забезпечення для передавання даних між приймачем та передавачем .....	22
2.2 Використання сенсора MPU6050 для відстеження руху .....	25
2.3 Підключення елементів пристрою .....	28
2.4 Розрахунок системи живлення .....	32
2.5 Алгоритм роботи пристрою .....	33
2.6 Комплементарний фільтр.....	36
2.7 Висновок до розділу 2 .....	38
3 РОЗРОБКА ПРОГРАМНОЇ ЧАСТИНИ .....	39
3.1 Середовище розробки.....	39
3.2 Налаштування Bluetooth модулів HC-05 в режимах Master та Slave за допомогою AT команд .....	40
3.3 User Story та Use Case діаграми для проекту .....	44

3.4 Програмне забезпечення для апаратної частини.....	47
3.5 Інтеграція периферійного пристрою з системами автоматизованого проєктування(3ds max) .....	55
3.6 Методи покращення периферійного пристрою для системи автоматизованого проєктування.....	58
3.7 Інтеграція пристрою в інших галузях.....	62
3.8 Висновок до розділу 3 .....	63
ВИСНОВКИ.....	65
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	66
ДОДАТОК А.....	68
ДОДАТОК Б .....	69
ДОДАТОК В.....	71
ДОДАТОК Г .....	74
ДОДАТОК Д.....	77



## ПЕРЕЛІК СКОРОЧЕНЬ

САПР	– Система автоматизованого проектування
СЗК	– Синдром зап'ястного каналу
CAD	– Система автоматизованого проектування
GFSK	– Gaussian Frequency Shift Keying
GND	– Ground
HID	– Human Interface Device
I2C	– Inter-Integrated Circuit
IDE	– Integrated Development Environment
ISM	– Industrial, Scientific, and Medical band
RX	– Receive
SCL	– Serial Clock Line
SDA	– Serial Data Line
SPI	– Serial Peripheral Interface
TX	– Transmit
UART	– Universal Asynchronous Receiver-Transmitter
USB	– Universal Serial Bus
VCC	– Voltage Common Collector
VIN	– Input Voltage

## ВСТУП

Сучасні системи автоматизованого проектування (CAD) відіграють ключову роль у розробці нових продуктів та конструкцій. Однак, разом із зростанням складності проектів і вимог користувачів, зростає і потреба у вдосконаленні периферійних пристроїв, які полегшують та прискорюють процес роботи. Відтак, розробка та впровадження нових периферійних пристроїв для систем CAD стає актуальною та важливою проблемою інженерної сфери.

Системи CAD активно використовують для розв'язання різноманітних завдань. Проте, щоб максимально використовувати потенціал цих систем, потрібні вдосконалені периферійні пристрої, які забезпечують швидке та зручне взаємодію з комп'ютером під час проектування.

**Метою кваліфікаційної роботи бакалавра** є розроблення периферійного пристрою, що сприятиме оптимізації процесу роботи в системах автоматизованого проектування.

**Об'єкт:** проектування моделей в системах автоматизованого проектування за допомогою периферійних пристроїв.

**Предмет:** апаратне забезпечення для розроблення периферійного пристрою, спрямований на поліпшення роботи в системах CAD.

Для досягнення поставленої мети необхідно вирішити такі **завдання:**

- Проаналізувати сучасні технології периферійних пристроїв для систем автоматизованого проектування;
- Провести огляд і аналіз існуючих аналогів та публікацій в галузі периферійних пристроїв для систем автоматизованого проектування;
- Визначити технічні вимоги та постановку задачі для розробки периферійного пристрою, враховуючи специфіку систем автоматизованого проектування;
- Провести моделювання та технічне проектування периферійного пристрою, включаючи функціональну/процесну модель, проектування інформаційного та програмно-технічного забезпечення;

– Здійснити програмну реалізацію периферійного пристрою та підготувати необхідну документацію, включаючи обґрунтування вибору програмних засобів та технологій, опис програмної реалізації та керівництво користувача.

**Практичне значення** отриманих результатів:

Даний пристрій можна використовувати як і для роботи з САД системами для моделювання та проектування, так і для домашнього використання, в якості периферійного пристрою в заміну класичним комп'ютерним мишам.

Апробація Горішна О. М. (бакалаврантка), Бурлаченко І. С. (ст. викладач каф. комп'ютерної інженерії, Чорноморський нац. ун-т ім. Петра Могили, м. Миколаїв, Україна). Інноваційні периферійні пристрої в комп'ютерних системах. «Могилянські читання – 2023: Досвід та тенденції розвитку суспільства в Україні: глобальний, національний та регіональний аспекти», присвячена Всесвітньому дню науки в ім'я миру та розвитку в рамках тижня науки з 6 по 10 листопада 2023 року.

# 1 АНАЛІТИЧНИЙ ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ ТА СКЛАДОВИХ СИСТЕМИ

## 1.1 Периферійні пристрої в САД

Системи автоматизованого проектування (САД) стали незамінним інструментом у різних галузях, таких як машинобудування, архітектура, дизайн та інженерія. Їхні можливості значно покращили процес проектування, але й викликали певні ергономічні проблеми, пов'язані з тривалим використанням традиційних комп'ютерних мишей.

Використання сучасних комп'ютерних мишей із традиційним дизайном може призводити до ряду проблем та незручностей, особливо при тривалому користуванні. Ці проблеми пов'язані з повторюваними рухами зап'ястя та кисті руки, що може призвести до синдрому зап'ястного каналу(СЗК).

СЗК – це поширене захворювання[1], яке характеризується болем, онімінням та поколюванням у зап'ясті та руці. СЗК виникає через стиснення серединного нерва в зап'ясті, що може бути спричинене повторюваними рухами, які робить рука при використанні миші.

З цього приводу вже було проведено ряд досліджень [2], що вказують на недоліки традиційної комп'ютерної миші, а також доводять факт того, що такі миші призводять ряду больових синдромів у зап'ясті та кисті руки.

Крім ергономічних проблем, використання традиційних мишей може негативно впливати на продуктивність та навчання в середовищі САД, особливо під час дистанційних занять. Під час онлайн-занять або використання онлайн-платформ для запису нотаток, користувачам часто доводиться писати від руки за допомогою миші. Це може бути повільніше та менш зручно, ніж використання ручки або стилуса на планшеті. Намагання писати мишкою може відбирати більше часу та зусиль, знижуючи ефективність навчання чи роботи.

Участь у вебінарах або онлайн-зустрічах часто потребує робити швидкі нотатки або виділяти важливу інформацію на екрані. Використання миші для

цих цілей може бути повільним та неточним, що може призвести до втрати важливої інформації.

Сучасні умови навчання та роботи характеризуються зростанням використання онлайн-платформ для різноманітних цілей. Ці платформи можуть включати електронні підручники, інтерактивні дошки, інструменти для спільної роботи та багато іншого. Взаємодія з цими платформами за допомогою миші може бути менш ефективною, ніж використання інших пристроїв, що дозволяють на більш природне керування та введення даних.

## 1.2 Наявні альтернативні рішення

Однак сучасними розробниками та виробниками пристроїв для комп'ютерів вже були розроблені альтернативні периферійні пристрої введення.

Одним із таких пристроїв є ергономічні безпроводні комп'ютерні миші[4]. Це комп'ютерні миші розроблені експертами з ергономіки з оптимальним кутом для комфортної роботи. Саме завдяки такому куту в  $57^\circ$  миша тримає зап'ястя піднятим, що зменшує тиск і забезпечує правильне положення передпліччя.



Рисунок 1.1 – Ергономічна безпроводна комп'ютерна миша

Проте у такої миші є й певні недоліки. Одним з них є висока вартість цього пристрою. Крім того, ця миша не адаптована для використання в системах САД, а лише вирішує проблему СЗК. Також на ринку представлені і інші альтернативні пристрої вводу, такі як: графічні планшети.

Графічний планшет (або дигітайзер) - це периферійний пристрій для введення цифрової інформації у графічному форматі. Він складається з електронного планшета з прямокутним меню та курсором. Графічний планшет має власну систему координат, і при руху курсора по планшету, координати передаються в комп'ютер. Ці дані можуть бути використані для створення об'єктів, ліній та полігонів у графічних програмах. Розміри планшетів варіюються від А6 до А0 формату.



Рисунок 1.2 – Графічний планшет

Графічні планшети [3] є гарним рішенням для САД, так як вони відзначаються високою точністю, що дозволяє художникам створювати деталізовані графічні роботи. Вони також мають ергономічний дизайн, який зменшує фізичні навантаження на руки та зап'ястя, що дозволяє працювати більше годин без відчуття дискомфорту.

Проте вартість графічних планшетів може бути високою, особливо для професійних моделей. Для щоденного використання графічний планшет не є найкращим рішенням. Також такий девайс є додатковим периферійним пристроєм, який не до кінця може замінити використання комп'ютерної миші.

### 1.3 Технологічне рішення

Щоб запропонувати технологічне рішення було розглянуто ряд можливих варіантів реалізації проєкту.

Одним із варіантів було розробка тачпаду на основі arduino micro та сенсорної панелі з використанням бібліотеки mouse.h, та допоміжних деталей: провідників та кнопок.



Рисунок 1.3 – Arduino Micro

Arduino Micro - це мініатюрна плата, що поєднує в собі всі необхідні можливості, простоту у використанні та низьку вартість. Вона може визначатися як периферійний пристрій. Саме завдяки тому, що у цьому цьому мікроконтролері використано ATmega32u4. Він має вбудовану підтримку з'єднання USB. Завдяки цьому Arduino Micro [5] може визначатися комп'ютером як послідовний COM порт, а й як периферійний пристрій. Можна сказати, що це зменшена копія Arduino Leonardo.

Проте такий варіант реалізації проекту має ряд недоліків пов'язаних з обмеження, які можуть вплинути на його ефективність та можливості використання, наприклад, використання кнопок тачпаду.



Рисунок 1.4 – Технічне рішення периферійного пристрою

Наступним варіантом, який було розглянуто – розробка тачпаду на основі двох ультразвукових датчиків та Arduino Uno.

Принцип роботи такого тачпаду полягав би в тому, що один із ультразвукових датчиків би відслідковував зміщення руху пальця користувача по координаті  $x$ , а інший – по координаті  $y$ .

Проте такий пристрій має великий ряд недоліків, пов'язаних із відслідковуванням руху пальця користувача, що зменшує ефективність такого пристрою для САД.

Наступний варіант технічного рішення – безпроводна комп'ютерна миша у виді пера. Для створення такого типу комп'ютерної миші потрібно вибрати компактний мікропроцесор з можливістю використання технології USB для розробки бездротової ручки-миші. Arduino Micro є ідеальним вибором для цієї задачі, оскільки має компактні розміри та вбудований USB. Він добре підходить для проектів, що потребують зовнішніх сенсорів та бездротового зв'язку.

Під час розробки такого пристрою також важливо обрати сенсор, який буде реагувати на рухи користувача. Основні критерії для вибору сенсора включають високу точність та швидкість реакції на рухи.

#### **1.4 Вибір комплектуючих**

Для компактності комп'ютерної миші у вигляді пера було прийнято рішення розробляти її із Bluetooth модулем. Тому цей проект буде містити два блоки: передавач і приймач. Блок передавача – це комп'ютерна миша, а блок приймача – зв'язок із комп'ютером.

Для спілкування між приймачем та передавачем потрібно підібрати два компактних Bluetooth модулі.



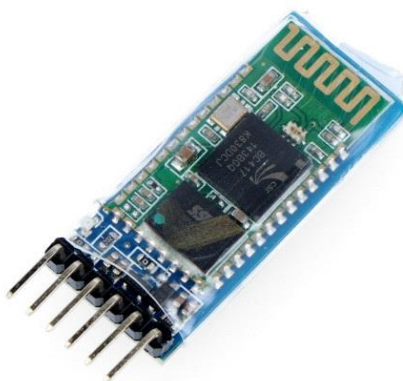


Рисунок 1.5 – Bluetooth-модуль HC-05

HC-05 - це популярний Bluetooth-модуль, який часто використовується для створення DIY-проектів, таких як Bluetooth-миші у вигляді пера. Він має ряд переваг, що робить його хорошим вибором для цієї задачі:

- HC-05 має простий інтерфейс команд, що робить його легким для програмування та використання з Arduino;
- HC-05 має компактний розмір, що робить його ідеальним для використання в невеликих проектах, таких як Bluetooth-миша у вигляді пера;
- HC-05 є порівняно недорогим модулем, що робить його доступним для DIY-проектів.
- Однак HC-05 також має деякі недоліки, які слід враховувати:
- HC-05 має діапазон дії близько 10 метрів, що може бути недостатнім для деяких проектів;
- HC-05 має низьку пропускну здатність, що може призвести до затримок при передачі даних;
- HC-05 не підтримує багато профілів Bluetooth, що може обмежити його функціональність.
- HC-05 є хорошим вибором для простих проектів, де важлива простота використання та низька вартість. RN-42 є кращим вибором для проектів, де потрібен більший діапазон дії, вища пропускну здатність або підтримка більшої кількості профілів Bluetooth.

Нижче наведено технічні характеристики даного модулю

Таблиця 1.1 – Технічні характеристики HC-05

Напруга живлення	3.3V - 5V
Робочий струм	< 50mA
Версія Bluetooth	2.0 + EDR
Діапазон передачі	до 10 метрів
Частота	2.4GHz ISM
Швидкість передачі даних	до 3 Мбіт/с
Підтримувані профілі	UART, SPP
Температурний діапазон	-20°C до +75°C
Розміри	37.5 мм x 16 мм x 4.5 мм

Також для даного проекту необхідно підібрати сенсор, котрий буде відповідати за позиціонування миші в просторі. В ході виконання кваліфікаційної роботи було розглянуто різні типи сенсорів, таких як: оптичні, акселерометри, гіроскопи.

Для даної роботи було розглянуто сенсор Pixart PMW3389.

Pixart PMW3389 - це високоточний оптичний сенсор, який часто використовується в ігрових мишах завдяки своїй надзвичайній точності та швидкості[6]. Цей сенсор забезпечує відмінне відстеження рухів та надзвичайно високу роздільну здатність, що дозволяє користувачам отримувати максимально точний контроль над курсором.

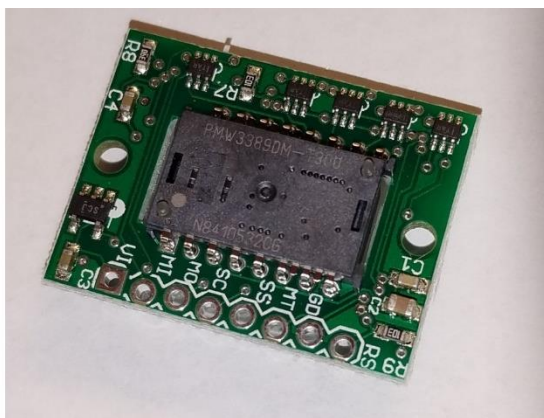


Рисунок 1.6 – Сенсор Pixart PMW3389

Особливості сенсора Pixart PMW3389 включають:

- Роздільна здатність до 16,000 DPI, що дозволяє адаптувати мишу під будь-які потреби користувача, від низької чутливості для точних рухів до високої для швидких дій;
- Висока швидкість обробки сигналу, що забезпечує обробляти до 450 дюймів в секунду (IPS) та витримувати прискорення до 50G, що забезпечує швидке і точне відстеження навіть при інтенсивних рухах;
- Завдяки швидкому зчитуванню даних та обробці сигналу, забезпечується миттєва реакція миші на рухи користувача, що є критично важливим в ігрових ситуаціях.

В таблиці 1.2 нижче наведено технічні характеристики даного сенсору.

Таблиця 1.2 – Технічні характеристики Pixart PMW3389

Максимальна роздільна здатність	16,000 DPI
Швидкість обробки сигналу	до 450 IPS
Максимальне прискорення	50G
Частота опитування	1000 Гц (1 мс)
Інтерфейс	SPI
Напруга живлення	3.3V
Споживання струму	10-15 мА
Температурний діапазон	-40°C до +85°C

Проте вартість через вартість даного датчику було прийнято рішення шукати більш оптимальний варіант для даного проекту, незважаючи на всі переваги цього сенсору.

Наступним сенсором, котрий було досліджено був акселерометр з гіроскопом – MPU6050.

- MPU6050 - це 6-осьовий датчик руху, який поєднує в собі 3-осьовий гіроскоп та 3-осьовий акселерометр. Він є одним з найпопулярніших датчиків руху для DIY-проектів, завдяки своїй простоті використання, точності та доступності.

- MPU6050 є найкращим рішенням для Bluetooth-миші у вигляді пера з наступних причин:
  - MPU6050 має високу точність, що робить його ідеальним для відстеження рухів пера з високою точністю;
  - MPU6050 має швидку частоту оновлення, що робить його ідеальним для проектів, де важлива швидка реакція;
  - MPU6050 має компактний розмір, що робить його ідеальним для використання в Bluetooth-миші у вигляді пера;
  - MPU6050 є порівняно недорогим датчиком, що робить його доступним для DIY-проектів.
  - MPU6050 легко використовувати з Arduino. Існує багато бібліотек Arduino, які полегшують роботу з MPU6050. Ці бібліотеки надають функції для читання даних датчика, калібрування датчика та обчислення кутів нахилу та повороту.

В таблиці 1.3 нижче наведено технічні характеристики сенсору MPU6050.

Таблиця 1.3 – Технічні характеристики – MPU6050

Напруга живлення	2.3V - 3.4V
Робочий струм	3.9 мА
Гіроскоп	±250, ±500, ±1000, ±2000 град/сек
Акселерометр	±2g, ±4g, ±8g, ±16g
Інтерфейс	I2C, SPI
Частота вибірки	до 1 кГц
Температурний діапазон	-40°C до +85°C
Вбудований цифровий датчик руху	6-осьовий
Розміри	4 мм х 4 мм х 0.9 мм

Зважаючи на ряд переваг, вартість та доступність було прийнято рішення використовувати в проекті саме цей сенсор, незважаючи на надчутливість та високий рівень шуму в даному сенсорі.

Також для передавача та приймача даних необхідно підібрати мікроконтролер. Для вибору мікроконтролера для передавача необхідно було враховувати кілька ключових критеріїв: розміри та компактність, підтримка HID-режиму (Human Interface Device), а також сумісність із бібліотекою <Mouse.h>, яка буде використовуватись у проекті. Крім того, важливо було забезпечити живлення сенсора MPU6050, який потребує 3,3V. З огляду на всі ці вимоги, Arduino Micro [5] виявився найкращим вибором.



Рисунок 1.7 – Arduino Micro

Arduino Micro - це компактна плата мікроконтролера, створена на базі ATmega32U4 у співпраці з Adafruit. Вона оснащена 20 цифровими входами/виходами, з яких 7 можуть використовуватися як ШІМ-виходи і 12 як аналогові входи. Плата також має кварцовий генератор на 16 МГц, роз'єм micro USB, заголовок ICSP і кнопку скидання. Все необхідне для підтримки мікроконтролера вбудовано, тому достатньо просто підключити плату до комп'ютера за допомогою кабелю micro USB, щоб почати роботу.

Таблиця 1.4 – Технічні характеристики Arduino Micro

Мікроконтролер	ATmega32U4
Напруга живлення	3.3V, 5V
Вхідна напруга (рекомендується)	7-12V
Цифрові I/O пінів	20

PWM пінів	7
Аналогові входи	12
Флеш пам'ять	32 КВ (4 КВ використовується для завантажувача)
SRAM	2.5 КВ
EEPROM	1 КВ
Тактова частота	16 МГц
Розміри	48 мм x 18 мм

Особливість Arduino Micro полягає в її здатності легко інтегруватися в макетні плати завдяки своєму зручному форм-фактору. Це робить її ідеальною для використання в компактних проектах. Подібно до Arduino Leonardo, плата використовує ATmega32U4, який має вбудовану підтримку USB-зв'язку. Це означає, що немає потреби у вторинному процесорі для забезпечення USB-з'єднання, що дозволяє мікроконтролеру з'являтися на підключеному комп'ютері як миша або клавіатура, а також як віртуальний (CDC) послідовний/SOM-порт.

Для вибору приймача вимог трохи менше і одна з найосновніших – підтримка HID-режиму, оскільки саме цей мікроконтролер буде підключено як USB HID пристрій. Також необхідно врахувати живлення для HC-05, який потребує 5V.

Для вибору мікроконтролера для приймача вимог значно менше, але одна з найважливіших — це підтримка HID-режиму, оскільки саме цей мікроконтролер буде підключено як USB HID пристрій. Крім того, необхідно врахувати живлення для модуля HC-05, який потребує 5V. З урахуванням цих вимог, Arduino Pro Micro є оптимальним вибором.



Рисунок 1.8 – Arduino Pro Micro

Arduino Pro Micro - це плата мікроконтролера, створена на основі мікроконтролера ATmega32U4 [5]. Вона була розроблена компанією Sparkfun Electronics і має відкриті вихідні коди, що включають схему і дизайн плати. Вона базується на архітектурі Arduino Leonardo, але має значно менший форм-фактор і вагу, що робить її ідеальною для проектів з обмеженим простором.

Таблиця 1.5 – Технічні характеристики Arduino Pro Micro

Мікроконтролер	ATmega32U4
Напруга живлення	5V
Вхідна напруга (рекомендується)	7-12V
Цифрові I/O пінів	12
PWM пінів	5
Аналогові входи	4
Флеш пам'ять	32 KB (4 KB використовується для завантажувача)
SRAM	2.5 KB
EEPROM	1 KB
Тактова частота	16 МГц (5V) або 8 МГц (3.3V)
Розміри	33 мм x 18 мм

Arduino Pro Micro має 22 контакти, з яких 4 є аналоговими входами (АЦП), а 5 можуть використовуватися як ШІМ-виходи (PWM). Плата також оснащена апаратним UART для передачі та прийому даних (Rx, Tx). Вона

працює на частоті 16 МГц і живиться від 5V. На вході RAW плата може приймати живлення до 10V, яке регулюється до 5V вбудованим регулятором.

### 1.5 Вимоги до апаратно- програмного комплексу

Апаратно-програмний комплекс для периферійного пристрою, який використовується в системах автоматизованого проектування (CAD), повинен включати ряд важливих компонентів та функцій для забезпечення ефективної та ергономічної роботи.

Таблиця 1.6 – Вимоги до апаратно- програмного комплексу

Arduino Micro	Мікроконтролер для передавача, підтримує HID-режим, забезпечує емуляцію миші
Arduino Pro Micro	Мікроконтролер для приймача, підтримує HID-режим, забезпечує емуляцію миші
MPU6050	Сенсор руху, вимірює кутові швидкості та акселерометричні дані
Bluetooth модулі HC-05	Забезпечують бездротовий зв'язок між передавачем і приймачем
Кнопки	Емуляція лівого та правого кліків миші
Подільники напруги	Захищають входні лінії Bluetooth модуля HC-05 від 5V сигналів
Елементи живлення	Забезпечують стабільне живлення для мікроконтролерів та інших компонентів

Для передавача необхідно використовувати мікроконтролер Arduino Micro, який базується на ATmega32U4. Цей мікроконтролер підтримує режим Human Interface Device (HID), що дозволяє емуляцію миші. Компактні розміри цього мікроконтролера дозволяють зручно інтегрувати його в пристрій. Arduino Micro також забезпечує достатню кількість цифрових входів для підключення сенсорів і кнопок.

Приймач має базуватись на мікроконтролері Arduino Pro Micro, який також використовує ATmega32U4 і забезпечує необхідну підтримку HID-режиму. Цей мікроконтролер має аналогічні характеристики та функціональні



можливості, що дозволяє використовувати його для прийому даних від передавача та емуляції миші.

Для визначення положення та руху пристрою використовується сенсор руху MPU6050. Цей сенсор забезпечує високоточне вимірювання кутових швидкостей та акселерометричних даних, що дозволяє точно контролювати рух курсора на екрані. MPU6050 легко інтегрується з Arduino через I2C інтерфейс, що робить його ідеальним вибором для цього проекту.

Для бездротового зв'язку між передавачем і приймачем використовуються Bluetooth модулі HC-05. Один модуль працює в режимі передавача, а інший – в режимі приймача. Це дозволяє передавати дані про рух і натискання кнопок бездротовим шляхом, забезпечуючи свободу руху та зручність використання.

Для емуляції лівого та правого кліків миші використовуються дві кнопки, підключені до відповідних цифрових входів мікроконтролера передавача. Це дозволяє користувачу здійснювати кліки так само, як при використанні традиційної миші.

Щоб забезпечити стабільне живлення для мікроконтролерів та інших компонентів, мікроконтролери можуть бути живлені через USB або від зовнішнього джерела живлення з відповідним регулюванням напруги. Для захисту входних ліній Bluetooth модуля HC-05, які працюють на 3.3V, від 5V сигналів від мікроконтролера, використовуються подільники напруги.

## **1.6 Висновки до розділу 1**

В даному розділі було детально проаналізовано проблематику використання традиційних комп'ютерних мишей у контексті систем автоматизованого проектування (CAD). Зокрема, були висвітлені ергономічні проблеми, пов'язані з тривалим використанням миші, можливість виникнення синдрому зап'ястного каналу в результаті повторюваних рухів, а також вплив цих аспектів на продуктивність та ефективність роботи чи навчання в середовищі CAD.

Також у даному розділі ретельно проаналізовано широкий спектр альтернативних рішень для вирішення проблем, пов'язаних з ефективністю та комфортом використання традиційних комп'ютерних мишей у сфері систем автоматизованого проектування. Було розглянуто різні інноваційні пристрої введення, включаючи трекболи, графічні планшети та спеціалізовані 3D-маніпулятори, які можуть запропонувати більш ергономічний та ефективний спосіб взаємодії з CAD-програмами.

У процесі аналізу було визначено, що важливими критеріями для вибору компонентів є їх сумісність із системами CAD, ергономічність, простота інтеграції та доступність на ринку. Було проаналізовано та підібрано компоненти для проекту, зокрема мікроконтролери, сенсори руху та бездротові модулі, які забезпечують необхідну функціональність та зручність використання.

Особливу увагу було приділено вибору мікроконтролера для передавача. Враховуючи вимоги щодо розмірів, компактності, підтримки HID-режиму, а також сумісності з бібліотекою ``Mouse.h``, було визначено, що Arduino Micro є найкращим вибором. Цей мікроконтролер, побудований на базі ATmega32U4, забезпечує необхідну підтримку HID та має компактні розміри, що робить його ідеальним для нашого проекту.

Для приймача було обрано Arduino Pro Micro, оскільки він також базується на ATmega32U4 і забезпечує необхідну функціональність та підтримку HID-режиму. Arduino Pro Micro має компактний форм-фактор і може забезпечити живлення для модуля HC-05, що робить його оптимальним вибором для інтеграції у проект.

Таким чином, у цьому розділі було не лише проаналізовано проблеми використання традиційних мишей у CAD-системах, але й підібрано відповідні компоненти для створення інноваційного периферійного пристрою, який покращує ергономічність та ефективність роботи користувачів у середовищі автоматизованого проектування.

## 2 РОЗРОБКА АПАРАТНОЇ ЧАСТИНИ

Щоб створити апаратну частину проекту, необхідно ретельно вивчити кожен компонент, щоб зрозуміти, як всі частини будуть взаємодіяти і функціонувати разом.

### 2.1 Апаратне забезпечення для передавання даних між приймачем та передавачем

Однією із складних частин кваліфікаційної роботи є налаштування зв'язку між двома Bluetooth- пристроями, які працюють в якості передавача та приймача даних. Щоб налаштувати роботу даних пристроїв спочатку необхідно розібратися із тим як влаштована передача даних через Bluetooth.

Bluetooth – це основна технологія бездротового зв'язку, яка використовується в нашому проекті для забезпечення передачі даних між компонентами системи [8]. Використання Bluetooth дозволяє уникнути необхідності фізичних з'єднань, що підвищує зручність та гнучкість пристрою.

На фізичному рівні модуль HC-05 використовує технологію радіочастотної модуляції для передачі даних. Основні аспекти фізичного рівня включають: частоту, модуляцію та синхронізацію.

Модуль HC-05 працює в діапазоні частот 2.4GHz ISM (Industrial, Scientific, and Medical band). Це стандартний частотний діапазон, використовуваний багатьма бездротовими пристроями, включаючи Wi-Fi та інші Bluetooth-пристрої. Використання цього діапазону забезпечує сумісність і стабільність зв'язку. Для зменшення інтерференції між пристроями, діапазон розбитий на 79 каналів, кожен шириною 1 МГц. Передача даних відбувається через постійне перемикання між цими каналами, що забезпечує надійність зв'язку і мінімізацію перешкод.

HC-05 використовує Gaussian Frequency Shift Keying (GFSK) для модуляції сигналу. GFSK є варіантом частотної модуляції, де бітові значення даних змінюють частоту носія. Наприклад, один біт може відповідати одній частоті, а нуль - іншій. Gaussian фільтр згладжує перехід між частотами, що

зменшує спектральну ширину сигналу і робить передачу більш стійкою до перешкод. Ця техніка дозволяє передавати дані на високих швидкостях, забезпечуючи при цьому низький рівень помилок.

Для ефективної передачі даних важливо, щоб передавач і приймач були синхронізовані. Синхронізація досягається через передачу спеціальних синхронізаційних сигналів. Передавач відправляє ці сигнали перед передачею основних даних, щоб приймач міг налаштуватися на правильну частоту і час. Цей процес включає як часову синхронізацію (встановлення правильних інтервалів для передачі бітів), так і частотну синхронізацію (забезпечення точного прийому частоти носія).

На каналному рівні HC-05 обробляє та керує передачею пакетів даних. Основні функції каналного рівня включають: поділ даних на пакети, контроль помилок та арбітраж доступу до каналу.

Дані, які передаються між пристроями, розбиваються на невеликі пакети. Кожен пакет включає заголовок, корисне навантаження (дані) і контрольну суму. Заголовок містить службову інформацію, таку як адреса приймача, номер пакету і тип пакету. Корисне навантаження містить безпосередньо дані, які передаються. Контрольна сума використовується для виявлення помилок при передачі.

Для забезпечення цілісності переданих даних використовуються методи виявлення та виправлення помилок. Одним з таких методів є Cyclic Redundancy Check (CRC). CRC додає до кожного пакету контрольну суму, обчислену на основі вмісту пакету. Приймач обчислює контрольну суму знову після отримання пакету і порівнює її з тією, що була передана. Якщо вони збігаються, пакет вважається коректним. Якщо ні - пакет відкидається і запитується повторна передача.

Frequency Hopping Spread Spectrum (FHSS) використовується для зменшення інтерференції і підвищення стійкості до перешкод. FHSS змінює частоту передачі на кожному пакеті за визначеним алгоритмом. Це означає, що навіть якщо один канал зазнає перешкод, інші канали можуть бути

вільними і використовуватися для передачі даних. FHSS також ускладнює перехоплення сигналу, що підвищує безпеку зв'язку.

Розглянемо процес передачі даних між передавачем та приймачем на сигнальному рівні.

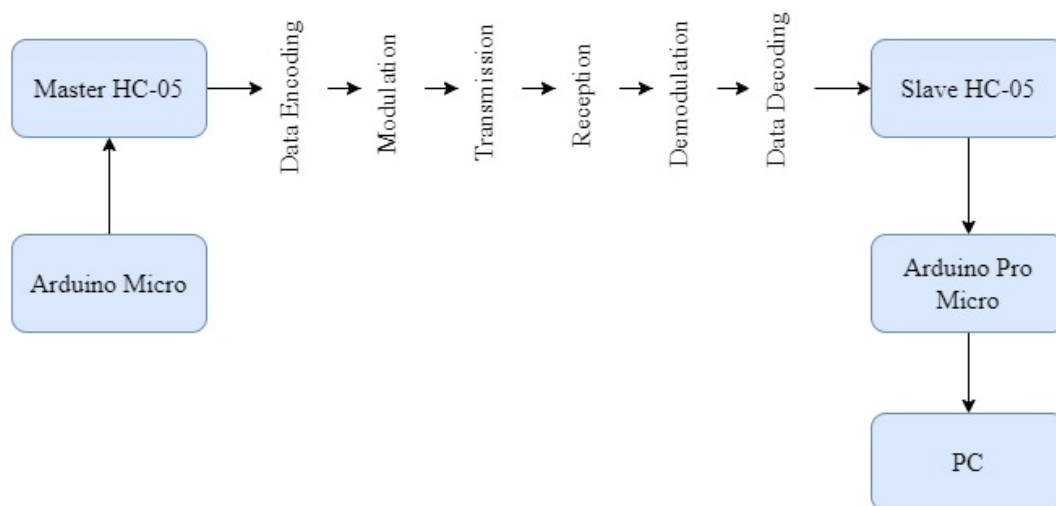


Рисунок 2.1 – Процес передачі даних між двома модулями HC-05

1. Перед передачею дані кодуються в бітову послідовність. Це включає додавання заголовків, контрольних сум та інших службових полів. Кодування може включати різні методи, такі як манчестерське кодування, яке забезпечує баланс між числом нулів і одиниць у послідовності, що полегшує синхронізацію.

2. Кодовані дані модуляції на радіочастотний сигнал за допомогою GFSK. Частота сигналу змінюється відповідно до бітової послідовності. Наприклад, для передачі "1" частота може збільшуватися, а для "0" зменшуватися. Це дозволяє передавати дані безперервним потоком, зменшуючи ймовірність помилок.

3. Модуль передає модульований сигнал через антену. Використовується FHSS для зменшення впливу інтерференції. Передача здійснюється в кілька етапів: передавач відправляє синхронізаційні сигнали, щоб приймач налаштувався на правильну частоту, а потім передає основні дані.

4. Приймач отримує сигнал через свою антену. Сигнал демодулюється, тобто перетворюється з радіочастотної форми назад у бітову послідовність. Це досягається за допомогою зворотного процесу модуляції. Приймач також перевіряє синхронізаційні сигнали для забезпечення коректного прийому даних.

5. Отримана бітова послідовність декодується для отримання початкових даних. Це включає перевірку контрольних сум для виявлення та виправлення помилок. Якщо контрольна сума не збігається, приймач запитує повторну передачу пакету. Декодування може також включати перевірку заголовків для визначення адресата і типу пакету.

6. Приймач і передавач синхронізуються на рівні частоти і часу передачі. FHSS забезпечує, що обидва пристрої змінюють канали у визначеній послідовності для уникнення колізій. Це підвищує надійність зв'язку і зменшує ймовірність втрати даних.

Ознайомившись із роботою модуля HC-05, необхідно детально розібратися з налаштуванням режимів на обох модулях та їх програмуванням. Ця інформація буде детально розглянута в третьому розділі кваліфікаційної роботи. Наступним кроком є вивчення принципу роботи сенсора та процесу передачі даних із сенсора MPU6050 на мікроконтролер.

## **2.2 Використання сенсора MPU6050 для відстеження руху**

MPU6050 - це інтегрований шестиосьовий інерційний сенсор, що поєднує в собі триосьовий гіроскоп і триосьовий акселерометр [7]. Він здатен вимірювати прискорення і кутові швидкості по трьох осях (X, Y, Z), що робить його ідеальним для додатків, які потребують точного відстеження руху.

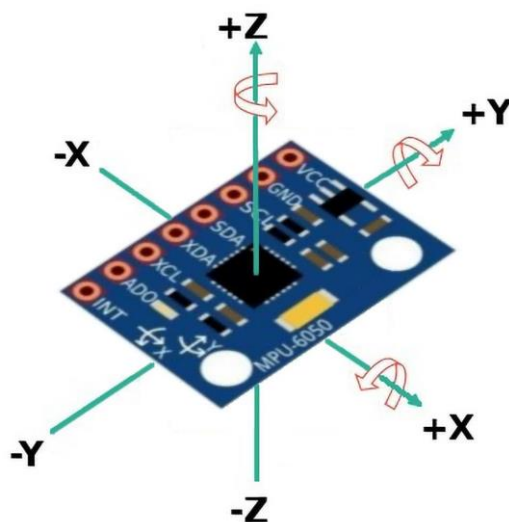


Рисунок 2.2 – MPU6050

MPU6050 складається з двох основних сенсорних елементів: акселерометра і гіроскопа.

### 2.2.1 Акселерометр

Акселерометр вимірює прискорення вздовж трьох осей (X, Y, Z). Прискорення - це зміна швидкості об'єкта з часом. Акселерометр здатний вимірювати як статичне прискорення (гравітація), так і динамічне прискорення (рух). Рівняння для обчислення результуючого прискорення виглядає так:

$$a = \sqrt{a_x^2 + a_y^2 + a_z^2}, \quad (2.1)$$

де  $a_x, a_y, a_z$  – виміряні прискорення вздовж осей X, Y і Z відповідно.

Акселерометри часто використовуються для визначення нахилу або орієнтації об'єкта, оскільки вони можуть виявляти силу гравітації.

### 2.2.2 Гіроскоп

Гіроскоп вимірює кутові швидкості або швидкість обертання вздовж трьох осей (X, Y, Z). Кутова швидкість - це зміна кута об'єкта з часом. Гіроскоп може виявляти швидкі і повільні обертання, що робить його ідеальним для відстеження рухів. Рівняння для обчислення результуючої кутової швидкості виглядає так:

$$\omega = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}, \quad (2.2)$$

де  $\omega_x, \omega_y, \omega_z$  – виміряні кутові швидкості вздовж осей X, Y і Z відповідно.

Гіроскопи часто використовуються для стабілізації рухомих об'єктів, таких як дрони, або для визначення швидкості обертання об'єктів.

MPU6050 використовує інтерфейс I2C або SPI для обміну даними з мікроконтролером.

### 2.2.3 Інтерфейси I2C та SPI

I2C - це двопровідний протокол, розроблений для забезпечення зв'язку між інтегральними схемами. Він використовується для передачі даних між мікроконтролером і периферійними пристроями, такими як сенсори, дисплеї та пам'яті [9].

Протокол I2C використовує лише дві лінії для передачі даних. Лінія SCL генерує тактові імпульси, які синхронізують передачу даних, а лінія SDA передає самі дані. Обидві лінії є відкритими колекторами (open-drain), тобто можуть бути з'єднані до живлення через підтягаючі резистори.

В I2C архітектурі один пристрій виступає як master (головний), який генерує тактові імпульси і ініціює зв'язок. Інші пристрої виступають як slave (ведені) і відповідають на запити master.

Кожен slave пристрій має унікальну 7- або 10-бітну адресу, яка використовується для ідентифікації при передачі даних.

Передача даних в I2C відбувається у вигляді байтів (8-біт), які супроводжуються додатковими бітом підтвердження (ACK/NACK). Загальна схема передачі даних включає стартовий сигнал (START), адресу пристрою, байти даних і стоп-сигнал (STOP).

SPI (Serial Peripheral Interface) - це інтерфейс для високошвидкісної передачі даних між мікроконтролером і периферійними пристроями [9]. SPI протокол використовує чотири основні лінії: SCK (Serial Clock), MOSI (Master Out Slave In), MISO (Master In Slave Out) і SS (Slave Select).



SPI протокол використовує чотири лінії для передачі даних. Лінія SCK генерує тактові імпульси, MOSI передає дані від master до slave, MISO передає дані від slave до master, а SS використовується для вибору slave пристрою.

SPI підтримує одночасну передачу даних в обох напрямках, що дозволяє здійснювати повнодуплексний зв'язок.

Як і в I2C, один пристрій виступає як master, який генерує тактові імпульси і керує зв'язком. Інші пристрої виступають як slave.

Дані передаються у вигляді байтів (8-біт) або більших фреймів. Кожен байт передається за один тактовий цикл SCK. Потік даних контролюється головним пристроєм (master), який генерує тактовий сигнал синхронізації (SCLK). Протягом одного тактового сигналу одночасно передається один біт даних від головного до підлеглого пристрою (по MOSI) та один біт від підлеглого до головного (по MISO).

I2C протокол забезпечує простоту та ефективність у багатопристроєвих системах завдяки своїй адресації і підтримці кількох slave пристроїв на одній шині. Проте, I2C має меншу швидкість передачі порівняно зі SPI. Проте в проєкті буде достатньо використання інтерфейсу I2C.

### **2.3 Підключення елементів пристрою**

Для забезпечення роботи сенсора MPU6050 в проєкті його потрібно підключити до мікроконтролера Arduino Micro за допомогою інтерфейсу I2C. Першим кроком у процесі підключення було забезпечення живлення сенсора. Контакт VCC сенсора MPU6050 було підключено до виходу 3.3V на Arduino Micro, що забезпечило сенсор необхідною напругою. Для забезпечення спільного заземлення контакт GND сенсора було з'єднано з контактом GND на Arduino Micro.

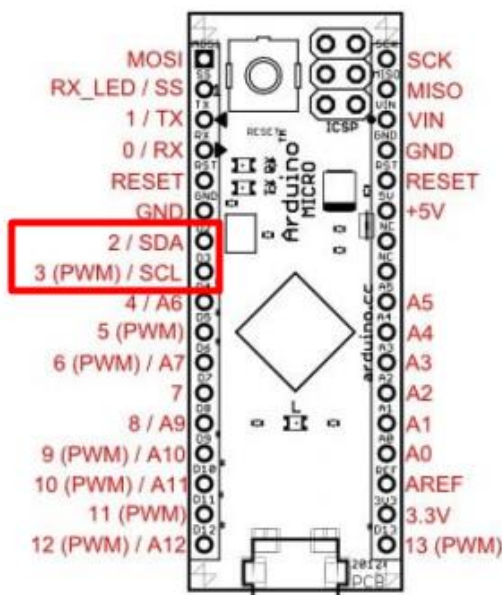


Рисунок 2.3 – Піни Arduino Micro

Наступним кроком було підключення ліній передачі даних. Лінія SCL (Serial Clock Line), яка відповідає за передачу тактових імпульсів, була підключена до контакту SCL на Arduino Micro. На цій платі цей контакт доступний через пін 3. Лінія SDA (Serial Data Line), яка використовується для передачі даних між сенсором і мікроконтролером, була підключена до контакту SDA на Arduino Micro, доступного через пін 2.

Таким чином, було забезпечено повне підключення сенсора MPU6050 до Arduino Micro, що дозволяє здійснювати обмін даними за допомогою інтерфейсу I2C.

Модуль HC-05 використовується для забезпечення бездротового зв'язку в проекті. Для його роботи необхідно було забезпечити правильне підключення до Arduino Micro. Спершу було підключено контакт VCC модуля HC-05 до виходу 5V на Arduino Micro, що забезпечило живлення модуля необхідною напругою. Контакт GND модуля було підключено до контакту GND на Arduino Micro для забезпечення спільного заземлення.

Далі було здійснено підключення контактів для передачі даних. Контакт TX (Transmit) модуля HC-05 було підключено до контакту RX (0) на Arduino Micro. Це з'єднання дозволяє передавати дані від модуля до мікроконтролера. Контакт RX (Receive) модуля HC-05 було підключено до контакту TX (1) на Arduino Micro через дільник напруги, оскільки RX модуля працює на 3.3V, а TX мікроконтролера видає 5V. Дільник напруги складався з резисторів 1 кОм і 2 кОм, що дозволило знизити напругу до безпечного рівня для модуля HC-05.

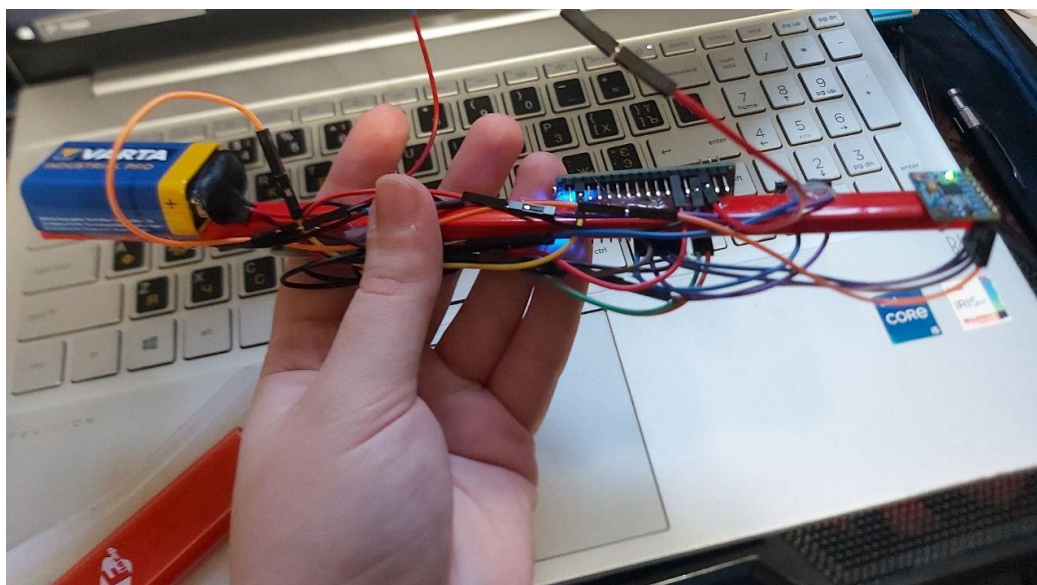


Рисунок 2.4 – Передавач пристрою

Процес підключення модуля HC-05 до Arduino Pro Micro аналогічний попередньому. Контакт VCC модуля HC-05 було підключено до виходу 5V на Arduino Pro Micro для забезпечення живлення. Контакт GND модуля було підключено до контакту GND на Arduino Pro Micro для спільного заземлення.

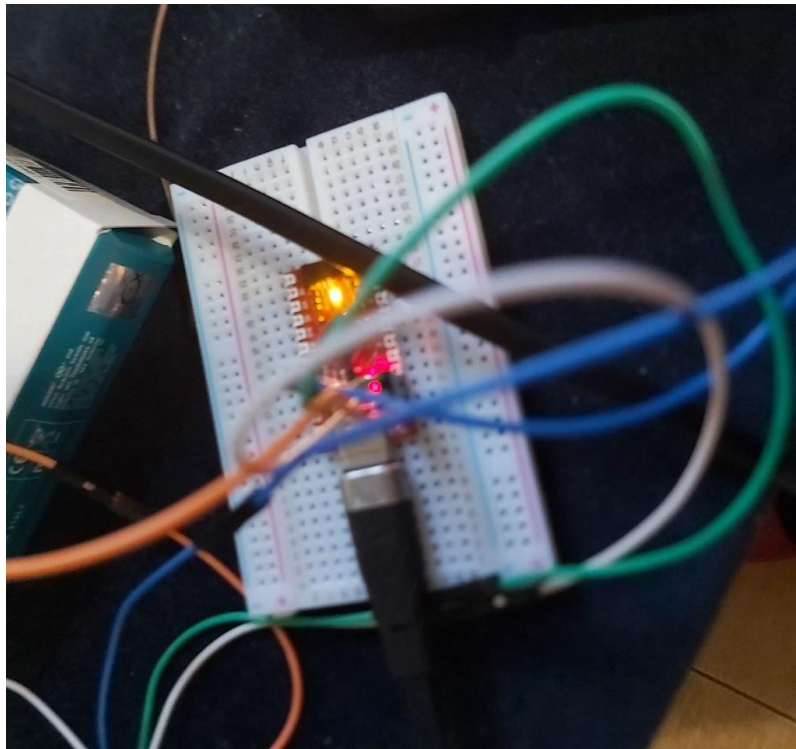


Рисунок 2.5 – Приймач пристрою

Для передачі даних контакт TX модуля HC-05 було підключено до контакту RX (0) на Arduino Pro Micro, а контакт RX модуля HC-05 було підключено до контакту TX (1) на Arduino Pro Micro через дільник напруги, щоб забезпечити безпечний рівень напруги для модуля.

Для забезпечення емуляції натискань миші в проекті використовуються дві кнопки. Ліва кнопка, яка відповідає за лівий клік миші, була підключена наступним чином: один контакт кнопки був з'єднаний з контактом GND на Arduino, а інший контакт був підключений до цифрового входу D2 на Arduino Micro та Arduino Pro Micro. Для забезпечення стабільного сигналу між контактом кнопки і контактом D2 було встановлено підтягуючий резистор номіналом 10 кОм, з'єднаний з контактом 5V.

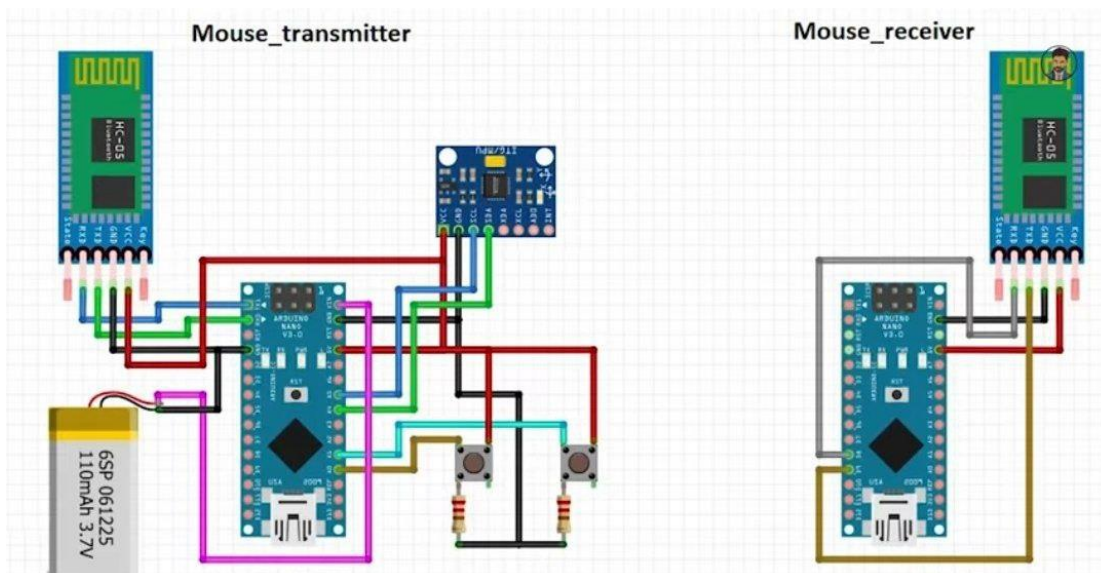


Рисунок 2.6 – Макетна схема периферійного пристрою

Права кнопка, що відповідає за правий клік миші, була підключена аналогічно: один контакт був з'єднаний з контактом GND, а інший - з цифровим входом D3 на Arduino Micro та Arduino Pro Micro. Також було встановлено підтягуючий резистор номіналом 10 кОм між контактом кнопки і контактом D3 для забезпечення стабільного сигналу.

Для забезпечення автономного живлення всіх компонентів в проекті використовувалась батарейка "Крона" (9V). Позитивний контакт батарейки був підключений до контакту Vin на Arduino Micro. Це дозволило використовувати вбудований регулятор напруги Arduino для зниження напруги до 5V, необхідних для живлення компонентів. Негативний контакт батарейки був з'єднаний з контактом GND на Arduino для забезпечення спільного заземлення.

## 2.4 Розрахунок системи живлення

Для забезпечення функціонування всієї системи необхідно правильно розрахувати систему живлення, враховуючи всі підключені компоненти. У даному проекті використовуються такі компоненти: Arduino Micro, сенсор MPU6050, модулі HC-05 та кнопки для емуляції кліків миші. Живлення для цих компонентів забезпечується за допомогою батарейки "Крона" (9V).

Щоб правильно розрахувати споживання енергії всією системою, необхідно визначити струм, споживаний кожним компонентом:

- Arduino Micro: Споживання струму становить приблизно 50 мА.
- MPU6050: Сенсор споживає близько 3.9 мА.
- HC-05: Модуль може споживати до 50 мА під час передачі даних.
- Кнопки: Споживання струму незначне, приблизно кілька мА в активному стані (приблизно 2 мА для двох кнопок).

Розрахунок загального струму споживання системи виглядає так:

$$I_{total} = 50\text{мА}(\text{ArduinoMicro}) + 3.9\text{мА}(\text{MPU6050}) + 50\text{мА}(\text{HC} - 05) \\ + 2\text{мА}(\text{кнопки}) = 105.9\text{мА}$$

Далі розглянемо алгоритм роботи пристрою.

## 2.5 Алгоритм роботи пристрою

Коли пристрій увімкнено, починається процес ініціалізації всіх підключених компонентів. Це включає запуск мікроконтролерів Arduino Micro та Arduino Pro Micro, які спершу виконують налаштування своїх внутрішніх ресурсів та підключених периферійних пристроїв.

На етапі ініціалізації Arduino Micro починає з налаштування MPU6050. Після підключення живлення до сенсора MPU6050, мікроконтролер встановлює початкові параметри сенсора через інтерфейс I2C. Це включає налаштування режимів вимірювання, діапазонів акселерометра та гіроскопа. Потім Arduino Micro перевіряє готовність сенсора до роботи, зчитуючи дані з його регістрів.



Рисунок 2.7 – Частина блок-схеми алгоритму роботи пристрою

Одночасно з цим, модуль HC-05 на Arduino Micro переходить у режим очікування з'єднання. Arduino Micro налаштовує UART інтерфейс для обміну даними з HC-05, встановлює швидкість передачі даних і перевіряє готовність модуля до роботи.

На іншому кінці, Arduino Pro Micro також починає свою ініціалізацію. Він налаштовує UART інтерфейс для зв'язку з HC-05 і перевіряє його готовність до встановлення з'єднання. Arduino Pro Micro готується до прийому даних від Arduino Micro і до передачі цих даних на комп'ютер через USB інтерфейс, емулюючи мишу.

Після завершення ініціалізації компонентів, модулі HC-05 на обох мікроконтролерах встановлюють бездротове з'єднання. HC-05 на Arduino Micro (ведучий) надсилає запит на підключення до HC-05 на Arduino Pro Micro (підлеглий). Як тільки підключення встановлено, обидва модулі синхронізуються для обміну даними.

Arduino Micro починає постійно зчитувати дані з сенсора MPU6050. Сенсор вимірює прискорення та кутові швидкості вздовж трьох осей (X, Y, Z). Дані передаються через I2C інтерфейс, де мікроконтролер зчитує їх з відповідних регістрів сенсора. Кожне вимірювання включає три значення прискорення ( $a_x, a_y, a_z$ ) та три значення кутової швидкості ( $\omega_x, \omega_y, \omega_z$ ).

Arduino Micro обробляє отримані дані для визначення рухів пристрою. Обробка включає фільтрацію шуму, обчислення кутів нахилу та обертання, а також перетворення цих даних у команди для управління курсором миші. Фільтрація може включати застосування алгоритмів, таких як Комплементарний фільтр або фільтр Калмана, для отримання більш точних даних про рух.

Оброблені дані передаються з Arduino Micro на Arduino Pro Micro через бездротовий зв'язок за допомогою модулів HC-05. Arduino Micro відправляє команди для управління курсором миші, включаючи інформацію про напрямок і швидкість руху, а також дані про натискання кнопок миші.

Arduino Pro Micro приймає дані, що надходять від HC-05, і передає їх на комп'ютер через USB інтерфейс. Програмне забезпечення на Arduino Pro Micro обробляє ці дані і виконує відповідні дії, емулюючи курсор миші. Це включає переміщення курсору на екрані та виконання кліків миші в залежності від отриманих команд.

Arduino Micro зчитує стан підключених кнопок, що імітують лівий і правий кліки миші. Коли користувач натискає кнопку, мікроконтролер визначає це як подію і відправляє відповідну команду через HC-05 на Arduino Pro Micro. Після отримання цієї команди, Arduino Pro Micro передає її на комп'ютер, де виконується відповідний клік миші.



Пристрій продовжує свою роботу, постійно зчитуючи дані з MPU6050, обробляючи їх, та передаючи результати на комп'ютер. Робота пристрою триває до тих пір, поки не вимкнеться живлення або не закінчиться заряд батарейки "Крона". У випадку розрядки батарейки або втрати з'єднання Bluetooth, система переходить у стан очікування або відновлення з'єднання.

## 2.6 Комплементарний фільтр

Комплементарний фільтр - це метод об'єднання сигналів з різних сенсорів для отримання більш надійного і точного результату [10]. Він базується на припущенні, що кожен сенсор має свої переваги в певних умовах, але також має недоліки. Об'єднуючи ці сигнали, можна зменшити вплив недоліків і підвищити точність результатів. У контексті інерційних вимірювальних одиниць (IMU), таких як MPU6050, комплементарний фільтр зазвичай об'єднує дані акселерометра і гіроскопа.

Основна ідея комплементарного фільтра полягає у використанні високочастотних компонентів одного сенсора (гіроскопа) і низькочастотних компонентів іншого сенсора (акселерометра). Це дозволяє компенсувати недоліки кожного з сенсорів. Гіроскоп надає точні короткострокові вимірювання, але з часом його дані схильні до дрейфу. Акселерометр, навпаки, забезпечує стабільні довгострокові вимірювання, але дуже чутливий до динамічних рухів і шуму.

Формула комплементарного фільтра виглядає так:

$$\theta(t) = \alpha(\theta(t-1) + \dot{\theta}(t)\Delta t) + (1 - \alpha)\theta_{acc}(t), \quad (2.3)$$

де  $\theta(t)$  – оцінений кут на поточний момент часу;

$\theta(t-1)$  – оцінений кут на попередній момент часу;

$\dot{\theta}(t)$  – кутова швидкість, виміряна гіроскопом;

$\Delta t$  – час між вимірюваннями;

$\theta_{acc}(t)$  – кут, виміряний акселерометром;

$\alpha$  – коефіцієнт фільтрації (зазвичай близький до 1).

Гіроскоп вимірює кутову швидкість, яку можна інтегрувати, щоб отримати кут. Проте інтеграція кутової швидкості призводить до накопичення похибки з часом, що викликає дрейф. Акселерометр вимірює гравітаційне прискорення, яке можна використовувати для обчислення нахилу. Проте акселерометр чутливий до динамічних прискорень і шуму.

Комплементарний фільтр використовує наступний підхід:

- $\alpha(\theta(t-1) + \dot{\theta}(t)\Delta t)$  - високочастотна компонента гіроскопа, що забезпечує точні короткострокові вимірювання.
- $(1 - \alpha)\theta_{acc}(t)$  - низькочастотна компонента акселерометра, що забезпечує стабільність довгострокових вимірювань.

Об'єднуючи ці компоненти, фільтр забезпечує стабільний і точний результат.

У даному проєкті комплементарний фільтр буде використовуватися для обробки даних з сенсора MPU6050. MPU6050 поєднує в собі триосьовий акселерометр і триосьовий гіроскоп, що дозволяє вимірювати як прискорення, так і кутову швидкість.

Використання комплементарного фільтра для обробки даних з MPU6050

1. Зчитуємо значення кутової швидкості з гіроскопа і прискорення з акселерометра.
2. Акселерометр вимірює компоненти прискорення  $a_x, a_y, a_z$ , які використовуються для обчислення кутів нахилу. Наприклад, кут нахилу  $\theta_{acc}$  можна обчислити як:

$$\theta_{acc} = \arctan\left(\frac{a_y}{a_z}\right), \quad (2.4)$$

3. Інтегруємо значення кутової швидкості для отримання зміни кута:

$$\theta_{acc} = \theta_{gyro} + \dot{\theta}(t)\Delta t, \quad (2.5)$$

4. Використовуємо комплементарний фільтр для комбінування даних з акселерометра та гіроскопа:

$$\theta(t) = \alpha(\theta_{gyro}) + (1 - \alpha)\theta_{acc}, \quad (2.6)$$

Комплементарний фільтр є ефективним інструментом для комбінування даних з різних сенсорів, таких як акселерометри і гіроскопи, для отримання більш точних результатів. У нашому проекті використання цього фільтра дозволить покращити точність визначення орієнтації та руху периферійного пристрою у вигляді пера, що значно підвищить зручність і ефективність роботи в середовищі автоматизованого проектування.

## **2.7 Висновок до розділу 2**

У даному розділі було детально розглянуто всі етапи підключення та налаштування апаратної частини проекту, що включає сенсор MPU6050, модулі HC-05, мікроконтролери Arduino Micro та Arduino Pro Micro, а також кнопки для емуляції кліків миші. Окрему увагу було приділено забезпеченню живлення системи та розрахунку споживання енергії для всіх компонентів.

Далі було налаштовано бездротове з'єднання між модулями HC-05 на Arduino Micro та Arduino Pro Micro, що забезпечує передачу оброблених даних про рух та команди для емуляції миші. Детально ознайомилися з принципами роботи Bluetooth-зв'язку, що дозволило створити стабільний канал зв'язку між мікроконтролерами за допомогою налаштування UART інтерфейсу.

Було розглянуто та розраховано систему живлення для даного пристрою.

Алгоритм роботи повністю готового пристрою включає ініціалізацію компонентів, встановлення бездротового з'єднання, зчитування та обробку даних з сенсора, передачу команд через Bluetooth, прийом даних на Arduino Pro Micro та передачу їх на комп'ютер, а також обробку натискань кнопок. Такий підхід забезпечує безперервну і точну взаємодію користувача з системою автоматизованого проектування.

## 3 РОЗРОБКА ПРОГРАМНОЇ ЧАСТИНИ

### 3.1 Середовище розробки

В даному проєкті використовуються плати Arduino Micro та Arduino Pro Micro, тому було прийнято рішення використовувати середовище розробки Arduino IDE .

Arduino IDE (Integrated Development Environment) є офіційним середовищем розробки для програмування та завантаження програмного забезпечення на платформи Arduino. Воно забезпечує користувачів зручним інтерфейсом для написання коду, його компіляції та завантаження на мікроконтролери Arduino.

Arduino IDE містить простий текстовий редактор для написання коду на мові програмування, яка базується на C/C++.

Середовище забезпечує компіляцію написаного коду в машинний код, який розуміє мікроконтролер. Інтегрований компілятор перевіряє код на наявність помилок і створює виконуваний файл.

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

Arduino IDE включає в себе велику кількість вбудованих бібліотек, які спрощують роботу з різними периферійними пристроями, такими як сенсори, модулі зв'язку та дисплеї. Крім того, середовище містить безліч прикладів коду, які можуть бути використані як основа для власних проєктів.

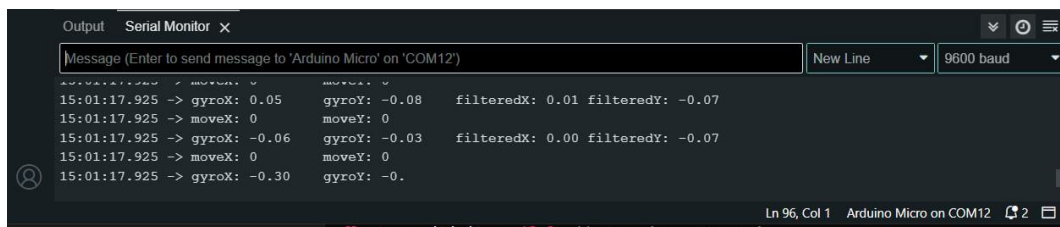


Рисунок 3.1 – Монітор послідовного порту з даними гіроскопу

Однією з важливих функцій Arduino IDE є монітор послідовного порту, який дозволяє відслідковувати дані, що надходять з мікроконтролера через послідовний порт. Це зручно для відлагодження і аналізу роботи програм.

### **3.2 Налаштування Bluetooth модулів HC-05 в режимах Master та Slave за допомогою AT команд**

Перед початком підключення всіх компонентів проєкту необхідно запрограмувати Bluetooth модулі HC-05 в режимах Master та Slave за допомогою AT команд.

Bluetooth модулі HC-05 можуть працювати у двох основних режимах: Master (ведучий) та Slave (підлеглий). Ці режими визначають, як модулі будуть взаємодіяти один з одним у бездротовій мережі.

Master (ведучий) – у цьому режимі ініціює з'єднання з іншими пристроями. Він виконує пошук доступних Bluetooth пристроїв і встановлює з'єднання з вибраним підлеглим пристроєм.

Slave (підлеглий) – у цьому режимі очікує на з'єднання від ведучого пристрою. Він не ініціює з'єднання, але готовий прийняти його, коли ведучий пристрій встановлює зв'язок.

Режими Master і Slave необхідні для організації бездротової мережі, де один пристрій виконує керівну функцію, а інший приймає команди та обмінюється даними.

Програмування модулів відбувається в режимі AT команд.

AT команди – це набір текстових команд, які використовуються для конфігурації та управління Bluetooth модулями [11]. Вони дозволяють змінювати налаштування модулів, такі як швидкість передачі даних, ім'я пристрою, роль (Master/Slave) тощо.

Основні AT команди:

Введення команди AT в терміналі дозволяє перевірити, чи модуль правильно підключений і працює. Після введення цієї команди модуль

повинен відповісти "ОК". Якщо відповіді немає, можливо, модуль не підключений або знаходиться не в режимі АТ команд.

Команда `AT+RESET` перезавантажує модуль. Це корисно для застосування нових налаштувань без фізичного відключення модуля від живлення. Після введення цієї команди модуль відповідає "ОК" і перезавантажується.

Команда `AT+VERSION?` дозволяє дізнатися версію програми керування модулем. Після введення цієї команди модуль відповідає версією програми, наприклад, "OKInvorV1.8".

Команда `AT+NAME=YourModuleName` встановлює ім'я модуля, яке буде відображатися під час пошуку Bluetooth пристроїв. Наприклад, `AT+NAME=MyBluetoothModule` змінить ім'я модуля на "MyBluetoothModule". Модуль відповідає "ОК", підтверджуючи зміну.

Команда `AT+ROLE=role` встановлює режим роботи модуля. Для встановлення режиму Master використовується `AT+ROLE=1`, а для режиму Slave - `AT+ROLE=0`. Після введення цієї команди модуль відповідає "ОК", підтверджуючи зміну ролі.

Команда `AT+PSWD=1234` змінює пароль для підключення до модуля. За замовчуванням пароль встановлений на "1234". Після введення цієї команди модуль відповідає "ОК", підтверджуючи зміну пароля.

Команда `AT+UART=baud,stop,parity` встановлює параметри UART. Наприклад, `AT+UART=9600,0,0` встановлює швидкість передачі даних на 9600 бод, один стоп-біт і відсутність парності. Модуль відповідає "ОК", підтверджуючи зміну налаштувань.

Команда `AT+CMODE=mode` встановлює режим з'єднання модуля. `AT+CMODE=0` означає, що модуль буде підключатися тільки до вказаного пристрою, а `AT+CMODE=1` дозволяє підключення до будь-якого доступного пристрою. Після введення цієї команди модуль відповідає "ОК".

Команда `AT+BIND=address` встановлює MAC-адресу іншого модуля для з'єднання. Адреса задається у форматі 0013:EF:000123. Наприклад,

AT+BIND=2015:88:040423. Після введення цієї команди модуль відповість "OK".

Команда AT+ADDR? дозволяє дізнатися MAC-адресу поточного модуля. Після введення цієї команди модуль відповість своєю MAC-адресою, наприклад, "OK+ADDR:2015:88:040423".

Розглянемо детально процес програмування модулів. Для початку позначимо модулі аби не переплутати який із них в якому режимі будемо програмувати.

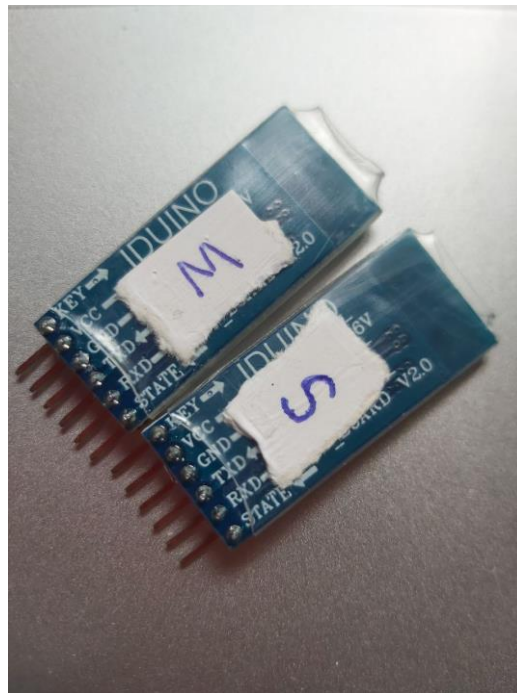


Рисунок 3.1 – Маркування Bluetooth модулів

Наступним кроком необхідно під'єднати один із модулів до Arduino Uno. Якщо на модулі присутня кнопка скидання, то затискаємо її та підключаємо до пінів плати.

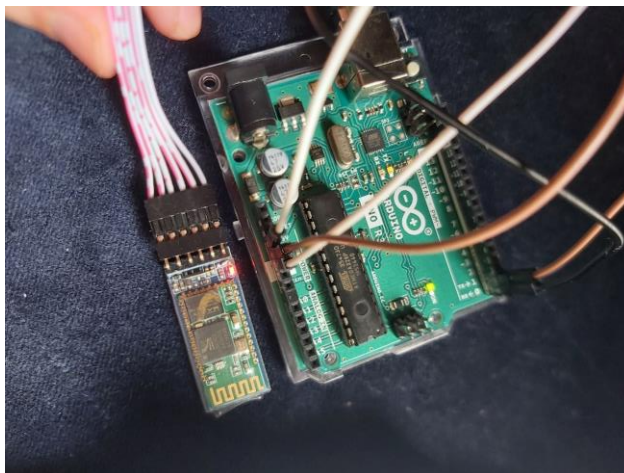


Рисунок 3.2 – Підключення Bluetooth модуля до Arduino Uno

За відсутності такої кнопки необхідно крім необхідних пінів підключити ще й пін KEУ до живлення 3.3V. Якщо все підключено правильно – світлодіод на модулі почне моргати з інтервалом в 2 секунди. Далі завантажимо простий код на Arduino Uno та почнемо налаштування модулів.

```
*sketch_settings_bluetooth: Блюют
Файл Редагування Формат Вигляд Довідка

#include <SoftwareSerial.h>
//
//ім'я                пароль                швидкість                CMODE                отримати адресу                записати адресу                роль
//запит                AT+NAME?                AT+PSWD?                AT+UART?                AT+CMODE?                AT+ADDR?                AT+BIND?                AT+ROLE?
//варіант відповіді  +NAME:SLAVE                +PIN:"1234"                +UART:9600,0,0                +CMODE:1                AT+ADDR?                AT+BIND=98d3,71,feaeff                +ROLE:0
//-----
//
//для слейва | AT+NAME=SLAVE | AT+PSWD="1234" | AT+UART=9600,0,0 | AT+CMODE=1 | +ADDR:98d3:71:feaeff | AT+BIND=98d3,71,feaeff | AT+ROLE=0
//для мастера | AT+NAME=MASTER | AT+PSWD="1234" | AT+UART=9600,0,0 | AT+CMODE=0 | AT+ADDR=98d3:71:feaeff | AT+BIND=98d3,71,feaeff | AT+ROLE=1

//скидання налаштувань AT+ORGL це для крайнього випадку, якщо щось напугали з налаштуваннями, і не можете знайти як виправити
#include <SoftwareSerial.h>

// використовує пини
const int arduino_rx = 5;
const int arduino_tx = 6;
SoftwareSerial mySerial (arduino_rx, arduino_tx);

void setup() {
  pinMode(arduino_rx, INPUT);
  pinMode(arduino_tx, OUTPUT);
  Serial.begin(9600); //в моніторі порта вибрати NL&CR, швидкість Arduino з комп'ютером 9600
  mySerial.begin(38400); // по замовчуванню швидкість Bluetooth модуля с Arduino в режимі AT команд 38400 іноді 9600
  Serial.println("<<< start! >>>");
  mySerial.println("AT");
}

void loop() {
  if (mySerial.available()) {
    char c = mySerial.read(); // читаємо з програмного Serial-порта
    Serial.print(c); // пишемо в штатний Serial-порт
  }
  if (Serial.available()) {
    char c = Serial.read(); // читаємо зі штатного Serial-порта
    mySerial.write(c); // пишемо в програмний Serial-порт
  }
}

//кінець скетчу
```

Рисунок 3.3 – Код для налаштування Bluetooth модулів

Виберемо на моніторі послідовного порту NL&CR, та змінюємо швидкість Arduino з комп'ютером на 9600. По замовчуванню швидкість Bluetooth модуля з Arduino в режимі AT команд 38400 іноді 9600.



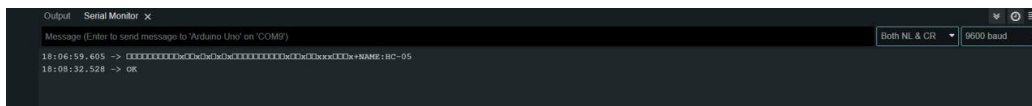


Рисунок 3.4 – Вибір параметрів в моніторі послідовного порту

Після налаштуємо для зручності імена пристроїв, пароль, швидкість, отримуємо адресу з підлеглого модуля та записуємо в адресу підключення до ведучого модуля. Також в самому кінці налаштуємо ролі модулів. Де значення 0 – це роль для підрядного модуля, а 1 – для ведучого. Якщо все зроблено вірно, то при підключенні двох модулів до живлення модулі з'єднаються одне з одним, однаково змінене моргання діоду на модулях буде свідчити про коректне підключення.

### 3.3 User Story та Use Case діаграми для проєкту

User Story (користувацька історія) – це короткий, простий опис функціональності програмного продукту, написаний з точки зору кінцевого користувача або клієнта. User Story зазвичай слідує формату "Як [роль], я хочу [функція], щоб [користь]", що допомагає зрозуміти вимоги користувачів та їх очікування.

User Story визначають потреби користувача, описують основні функціональності, які потрібно реалізувати. Також User Story визначають критерії прийняття, що дозволяє оцінити чи відповідає розроблена функціональність потребам користувача.

Use Case Діаграма – це графічне представлення взаємодій між користувачами (або іншими системами) та системою, яка розробляється. Діаграми Use Case описують різні способи використання системи та визначають, хто буде взаємодіяти із системою і які функції вона повинна виконувати.

Для даного проєкту User Story будуть виглядати наступним чином:

– Назва: «використання бездротового пристрою для керування мишею в системах автоматизованого проектування (САПР)»;

– Опис: як користувач, я хочу використовувати бездротовий пристрій у вигляді пера для керування мишею на комп'ютері під час роботи з САПР, щоб зменшити фізичне навантаження та підвищити точність і зручність роботи.

Критерії прийняття:

1. Пристрій повинен коректно зчитувати рухи пера та передавати їх на комп'ютер;
2. Передача даних повинна бути бездротовою і надійною;
3. Пристрій повинен мати дві кнопки для емуляції лівого та правого кліків миші;
4. Дані про рух та натискання кнопок повинні коректно відображатися на комп'ютері у вигляді переміщення курсора та виконання кліків.

У даному проекті ми будемо використовувати бездротовий пристрій для керування мишею на комп'ютері, який складається з двох основних компонентів: передавача та приймача. Передавач оснащений сенсором MPU6050, який вимірює рухи користувача, та кнопками для емуляції миші. Приймач отримує дані від передавача через Bluetooth та передає їх на комп'ютер через USB інтерфейс, емулюючи мишу. Нижче наведено детальний опис процесу та відповідну Use Case діаграму.

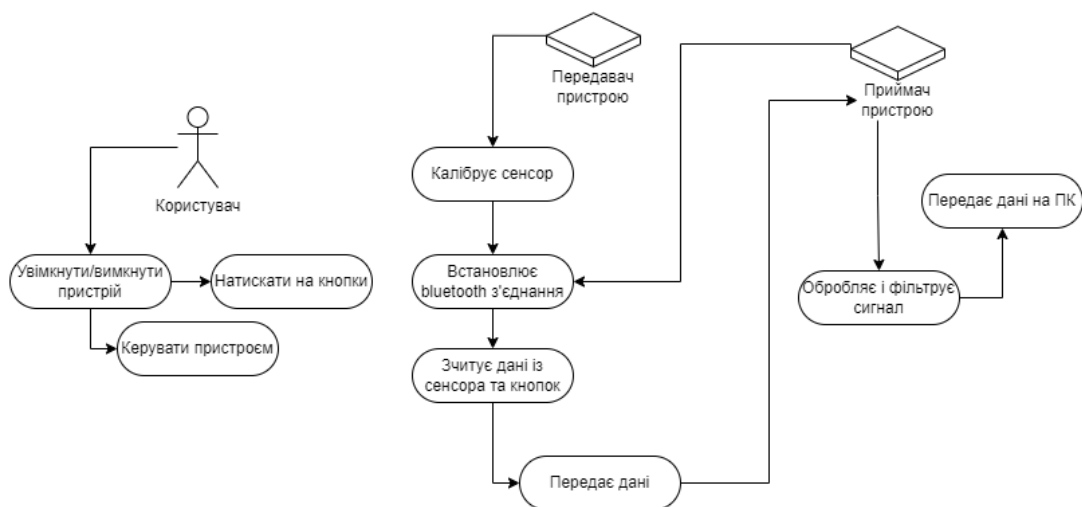


Рисунок 3.5 – Use Case діаграма периферійного пристрою

Користувач підключає передавач до джерела живлення, такого як батарейка "Крона", яка забезпечує необхідну напругу для роботи сенсора MPU6050 і мікроконтролера Arduino Micro. Приймач підключається до комп'ютера через USB, що забезпечує живлення для Arduino Pro Micro і дозволяє передачу даних на комп'ютер.

Після підключення пристрою користувач вмикає його. Мікроконтролери Arduino Micro та Arduino Pro Micro виконують ініціалізацію, налаштовуючи свої внутрішні ресурси. Це включає запуск сенсора MPU6050, налаштування режимів вимірювання, діапазонів акселерометра та гіроскопа. Мікроконтролери також налаштовують UART інтерфейс для обміну даними з модулями HC-05.

Після увімкнення пристрою передавач автоматично калібрує сенсор MPU6050 для точного вимірювання рухів. Калібрування включає зчитування даних з гіроскопа протягом певного часу для визначення середніх значень, які використовуються для корекції подальших вимірювань. Це дозволяє зменшити похибки та забезпечити більш точні результати.

Користувач налаштовує Bluetooth модулі для встановлення бездротового з'єднання між передавачем і приймачем. Для цього модулі HC-05 потрібно налаштувати у відповідні режими Master і Slave за допомогою AT команд. Модуль на передавачі налаштовується в режим Master, а модуль на приймачі в режим Slave. Після налаштування модулів вони встановлюють з'єднання між собою.

Після успішного встановлення з'єднання передавач зчитує дані з сенсора MPU6050 та кнопок, обробляє їх і передає на приймач через Bluetooth. Дані включають інформацію про кутові швидкості та стан кнопок. Наприклад, рухи по осям X і Y передаються у вигляді числових значень, які представляють собою швидкість обертання.

Приймач отримує дані від передавача через Bluetooth модуль HC-05. Потім він обробляє ці дані для визначення руху курсора і стану кнопок, та передає їх на комп'ютер через USB інтерфейс, емулюючи мишу. Приймач

виконує аналіз отриманих даних, обчислює переміщення курсора на екрані та визначає, чи були натиснуті кнопки миші.

Користувач керує курсором миші та виконує кліки, використовуючи передавач у вигляді пера. Комп'ютер реагує на дані, отримані від приймача, і виконує відповідні дії, такі як переміщення курсора або натискання кнопок миші. Це дозволяє користувачеві інтуїтивно та ефективно взаємодіяти з системою автоматизованого проектування, знижуючи фізичне навантаження і підвищуючи точність роботи.

Таким чином, весь процес включає підключення та ініціалізацію пристрою, калібрування сенсора, встановлення Bluetooth з'єднання, передачу даних від передавача до приймача, обробку даних на приймачі та виконання дій мишею на комп'ютері. Це забезпечує зручне та ефективне управління курсором миші у середовищі САПР.

### **3.4 Програмне забезпечення для апаратної частини**

У попередньому розділі було розглянуто основні моменти взаємодії користувача із системою за допомогою Use Case діаграм. Це дозволило нам детально описати процеси, які відбуваються під час роботи пристрою, включаючи ініціалізацію, калібрування сенсора, встановлення Bluetooth з'єднання, передачу та обробку даних, а також емуляцію миші на комп'ютері. Тепер ми можемо перейти до проектування програмного забезпечення для апаратної частини системи, яке забезпечить реалізацію цих процесів.

Програмне забезпечення для апаратної частини складається з двох основних компонентів: програмного коду для передавача та програмного коду для приймача. Передавач використовує мікроконтролер Arduino Micro та сенсор MPU6050 для зчитування рухів і кнопок, а також модуль Bluetooth HC-05 для передачі цих даних. Приймач, на основі мікроконтролера Arduino Pro Micro, отримує дані від передавача через Bluetooth, обробляє їх та передає на комп'ютер для емуляції миші.

### 3.4.1 Ініціалізація компонентів

Ініціалізація компонентів передбачає налаштування сенсора MPU6050, модулів Bluetooth та пінів для кнопок.

```
#include <Wire.h>
#include <I2Cdev.h>
#include <MPU6050.h>

MPU6050 mpu;

const int leftButtonPin = 4; // Пін для лівої кнопки миші
const int rightButtonPin = 5; // Пін для правої кнопки миші

void setup() {
  Serial.begin(9600); // Ініціалізація серійного зв'язку для відлагодження
  Serial1.begin(9600); // Ініціалізація апаратного серійного зв'язку для Bluetooth
```

a)

```
Wire.begin();
pinMode(leftButtonPin, INPUT_PULLUP);
pinMode(rightButtonPin, INPUT_PULLUP);

mpu.initialize();
if (!mpu.testConnection()) {
  Serial.println("MPU6050 connection failed");
  while (1);
}

calibrateGyroscope();
Serial.println("Setup complete");
}
```

б)

Рисунок 3.6 – Ініціалізація компонентів (а, б)

Спочатку імпортуються необхідні бібліотеки для роботи з сенсором MPU6050 та протоколом I2C. Використовується бібліотека Wire.h для реалізації I2C зв'язку, бібліотека I2Cdev.h для зручної роботи з I2C пристроями та бібліотека MPU6050.h для взаємодії з сенсором.

Далі відбувається ініціалізація серійного зв'язку для відлагодження за допомогою Serial.begin(9600) та апаратного серійного зв'язку для Bluetooth за

допомогою `Serial1.begin(9600)`. Це дозволяє передавати дані на комп'ютер та отримувати їх через Bluetooth модуль.

Наступний крок - ініціалізація I2C інтерфейсу за допомогою `Wire.begin()`, що дозволяє комунікувати з сенсором MPU6050. Встановлення пінів для кнопок здійснюється командами `pinMode(leftButtonPin, INPUT_PULLUP)` та `pinMode(rightButtonPin, INPUT_PULLUP)`, які налаштовують піни для кнопок на вхід з внутрішніми підтягуючими резисторами.

Далі відбувається ініціалізація сенсора MPU6050 за допомогою методу `mpu.initialize()`. Після цього перевіряється успішність підключення сенсора за допомогою `mpu.testConnection()`. Якщо підключення не вдалося, виводиться повідомлення "MPU6050 connection failed" і виконання програми зупиняється.

### 3.4.2 Калібрування сенсора

Калібрування сенсора необхідне для точного вимірювання рухів. Калібрування передбачає зчитування даних з гіроскопа протягом певного часу для визначення середніх значень, які використовуються для корекції подальших вимірювань. Це дозволяє зменшити похибки і забезпечити точні результати.

```
void calibrateGyroscope() {
    float sumX = 0, sumY = 0;
    int numSamples = 100;

    for (int i = 0; i < numSamples; i++) {
        int16_t gx, gy, gz;
        mpu.getRotation(&gx, &gy, &gz);
        sumX += gx / 131.0;
        sumY += gy / 131.0;
        delay(10);
    }
}
```

a)

```
gyroXOffset = sumX / numSamples;  
gyroYOffset = sumY / numSamples;  
  
Serial.print("gyroXOffset: ");  
Serial.println(gyroXOffset);  
Serial.print("gyroYOffset: ");  
Serial.println(gyroYOffset);  
}
```

б)

Рисунок 3.7 – Калібрування сенсора (а, б)

Далі буде зчитування даних з гіроскопа 100 разів. Дані з гіроскопа зчитуються за допомогою методу `mpu.getRotation(&gx, &gy, &gz)`, де `gx`, `gy` і `gz` - це значення обертання по осям X, Y і Z відповідно. Отримані значення накопичуються у змінних `sumX` і `sumY`. Після цього обчислюються середні значення для кожної осі, які використовуються як зсуви для корекції подальших вимірювань. Значення зсувів виводяться на серійний монітор для відлагодження.

### 3.4.3 Зчитування даних з сенсора та кнопок

Після калібрування сенсора відбувається зчитування даних з сенсора та кнопок у головному циклі програми. Отримані дані фільтруються та обробляються для зменшення шуму і забезпечення стабільної роботи пристрою.

```
void loop() {  
  int16_t gx, gy, gz;  
  mpu.getRotation(&gx, &gy, &gz); // Отримання даних з гіроскопа  
  
  float gyroX = gx / 131.0 - gyroXOffset;  
  float gyroY = gy / 131.0 - gyroYOffset;  
  
  static float filteredX = 0;  
  static float filteredY = 0;
```

а)

```
// Згладжування сигналу
filteredX = alpha * gyroX + (1 - alpha) * filteredX;
filteredY = alpha * gyroY + (1 - alpha) * filteredY;

int leftButtonState = digitalRead(leftButtonPin);
int rightButtonState = digitalRead(rightButtonPin);
```

б)

```
// Надсилання даних через блютуз
Serial1.print(filteredX);
Serial1.print(",");
Serial1.print(filteredY);
Serial1.print(",");
Serial1.print(leftButtonState);
Serial1.print(",");
Serial1.println(rightButtonState);

delay(5); // Невелика затримка для стабільності
}
```

в)

Рисунок 3.8 – Зчитування даних з сенсора та кнопок (а, б, в)

Зчитуємо значення обертання по осям X, Y і Z за допомогою методу `mpu.getRotation(&gx, &gy, &gz)`. Отримані значення обертання нормалізуються і коригуються за допомогою зсувів, отриманих під час калібрування. Для зменшення шуму дані фільтруються за допомогою алгоритму згладжування сигналу. Стан кнопок зчитується за допомогою `digitalRead(leftButtonPin)` та `digitalRead(rightButtonPin)`.

Отримані та оброблені дані передаються на приймач через модуль Bluetooth за допомогою `Serial1.print()`. Затримка у 5 мілісекунд додається для стабільності роботи системи.



### 3.4.4 Ініціалізація компонентів для приймача

```
#include <Mouse.h>

void setup() {
  Serial.begin(9600);
  Serial1.begin(9600); // Ініціалізація апаратного серійного зв'язку для Bluetooth
  Mouse.begin();
  Serial.println("Setup complete");
}
```

Рисунок 3.9 – Ініціалізація компонентів для приймача

Ініціалізація компонентів приймача включає налаштування серійного зв'язку для відлагодження та апаратного серійного зв'язку для Bluetooth модуля. Також ініціалізується бібліотека Mouse, яка дозволяє емулювати мишу на комп'ютері.

### 3.4.5 Прийом даних через Bluetooth

Прийом даних через Bluetooth здійснюється в головному циклі програми. Приймач постійно зчитує дані, що надходять від передавача через модуль Bluetooth HC-05.

```
void loop() {
  if (Serial1.available()) {
    // Зчитування даних з Bluetooth
    String data = Serial1.readStringUntil('\n');
    Serial.println("Received data: " + data); // Відображення прийнятих даних

    // Розбір даних
    float gyroX = data.substring(0, data.indexOf(',')).toFloat();
    data = data.substring(data.indexOf(',') + 1);
    float gyroY = data.substring(0, data.indexOf(',')).toFloat();
    data = data.substring(data.indexOf(',') + 1);
    int leftButtonState = data.substring(0, data.indexOf(',')).toInt();
    int rightButtonState = data.substring(data.indexOf(',') + 1).toInt();
  }
}
```

a)

```
Serial.print("gyroX: ");  
Serial.print(gyroX);  
Serial.print(" gyroY: ");  
Serial.print(gyroY);  
Serial.print(" leftButtonState: ");  
Serial.print(leftButtonState);  
Serial.print(" rightButtonState: ");  
Serial.println(rightButtonState);
```

б)

```
// Обчислення руху миші  
int moveX = baseSensitivity * speedFactor * gyroX;  
int moveY = baseSensitivity * speedFactor * gyroY;  
  
if (abs(moveX) < 1) moveX = 0;  
if (abs(moveY) < 1) moveY = 0;  
  
Serial.print("moveX: ");  
Serial.print(moveX);  
Serial.print(" moveY: ");  
Serial.println(moveY);
```

в)

```
// Рух миші  
Mouse.move(moveX, moveY);  
  
// Обробка натискань кнопок миші  
if (leftButtonState == LOW) {  
    Mouse.press(MOUSE_LEFT);  
} else {  
    Mouse.release(MOUSE_LEFT);  
}
```

г)

```
if (rightButtonState == LOW) {  
    Mouse.press(MOUSE_RIGHT);  
} else {  
    Mouse.release(MOUSE_RIGHT);  
}  
  
delay(10); // Невелика затримка для стабільності  
}
```

д)

Рисунок 3.10 – Прийом даних через Bluetooth (а, б, в, г, д)

Приймач постійно перевіряє, чи доступні дані з Bluetooth. Коли дані стають доступними, вони зчитуються за допомогою `Serial1.readStringUntil('\n')` і виводяться на серійний монітор для відлагодження.

Приймач розбирає отримані дані, розділяючи їх на компоненти: кутові швидкості по осям X та Y, а також стан кнопок. Для цього використовуються методи `substring()` і `toFloat()` для числових даних та `toInt()` для стану кнопок.

Отримані дані про кутові швидкості обробляються для визначення переміщення курсора миші. Застосовується базова чутливість і фактор швидкості для обчислення значень `moveX` та `moveY`. Для зменшення дрібних коливань застосовується порогове значення, нижче якого переміщення не враховується.

Після обробки даних відбувається емуляція миші. Метод `Mouse.move()` використовується для переміщення курсора миші відповідно до обчислених значень `moveX` та `moveY`. Стан кнопок визначає, чи потрібно виконати клік миші. Якщо кнопка натиснута, використовується метод `Mouse.press()`, якщо відпущена - `Mouse.release()`.

### **3.5 Інтеграція периферійного пристрою з системами автоматизованого проєктування(3ds max)**

Інтеграція периферійного пристрою з системами автоматизованого проєктування (САПР) є важливою темою, оскільки вона дозволяє покращити ефективність та зручність роботи дизайнерів та інженерів, які використовують такі системи. Розуміння того, як використовувати новітні технології для підвищення продуктивності, є ключовим аспектом у розвитку інноваційних рішень для САПР. У даному підрозділі розглянемо інтеграцію периферійного пристрою з однією з найбільш популярних програм для 3D моделювання та анімації - 3ds Max.

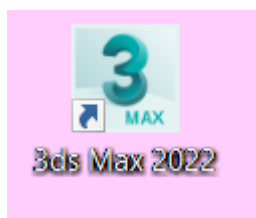


Рисунок 3.11 – 3ds Max

3ds Max - це потужне програмне забезпечення для 3D моделювання, анімації, візуалізації та рендерингу, розроблене компанією Autodesk. Ця програма широко використовується в різних галузях, таких як архітектурне проєктування, ігрова індустрія, кінематографія та візуальні ефекти. 3ds Max надає користувачам безліч інструментів для створення складних тривимірних моделей, анімаційних сцен і високоякісних рендерів.

Інтеграція периферійного пристрою, оснащеного сенсором MPU6050 та модулями HC-05, з 3ds Max може значно покращити зручність використання цього програмного забезпечення. Використання бездротового пера для керування мишею дозволить користувачам легко маніпулювати 3D об'єктами, здійснювати точні переміщення, обертання та масштабування.

Інтеграція периферійного пристрою, оснащеного сенсором MPU6050 та модулями HC-05, з 3ds Max може значно покращити зручність використання цього програмного забезпечення. Використання бездротового пера для

керування мишею дозволить користувачам легко маніпулювати 3D об'єктами, здійснювати точні переміщення, обертання та масштабування.

```
void loop() {  
  int leftButtonState = digitalRead(leftButtonPin);  
  int rightButtonState = digitalRead(rightButtonPin);  
  
  // Обробка довгих натискань  
  if (leftButtonState == LOW) {  
    if (leftButtonPresTime == 0) {  
      leftButtonPresTime = millis();  
    } else if (millis() - leftButtonPresTime > longPressDuration) {  
      activateHotkey("left");  
      leftButtonPresTime = 0; // Скидання часу натискання  
    }  
  }  
}
```

а)

```
} else {  
  leftButtonPresTime = 0;  
}  
  
if (rightButtonState == LOW) {  
  if (rightButtonPresTime == 0) {  
    rightButtonPresTime = millis();  
  } else if (millis() - rightButtonPresTime > longPressDuration) {  
    activateHotkey("right");  
    rightButtonPresTime = 0; // Скидання часу натискання  
  }  
}
```

б)

```
} else {  
  rightButtonPresTime = 0;  
}  
  
delay(10); // Невелика затримка для стабільності  
}
```

в)

Рисунок 3.12 – Зчитування даних з сенсора та кнопок (а, б, в)

Функція loop() в Arduino виконується безперервно після завершення функції setup(). У даному випадку loop() обробляє натискання кнопок миші та активує відповідні гарячі клавіші, якщо кнопки натиснуті достатньо довго.

Спершу зчитуються стани лівої та правої кнопок миші за допомогою функції `digitalRead()`. Змінні `leftButtonState` та `rightButtonState` отримують значення `LOW`, якщо відповідні кнопки натиснуті, і `HIGH`, якщо кнопки не натиснуті. Це дозволяє визначити, чи кнопка натиснута в даний момент.

Далі, код перевіряє стан лівої кнопки миші. Якщо ліва кнопка натиснута (`leftButtonState` дорівнює `LOW`), то перевіряється, чи час натискання (`leftButtonPressTime`) дорівнює нулю. Якщо так, то це означає, що кнопка тільки що була натиснута, і зберігається поточний час за допомогою функції `millis()`, яка повертає час у мілісекундах з моменту запуску програми.

Якщо кнопка натиснута протягом деякого часу, наступна перевірка визначає, чи натискання тривало більше заданої тривалості для довгого натискання (`longPressDuration`). Якщо час, що минув з моменту першого натискання, перевищує `longPressDuration`, викликається функція `activateHotkey("left")`, яка активує гарячу клавішу для лівої кнопки. Після активації гарячої клавіші змінна `leftButtonPressTime` скидається до нуля, що означає, що цикл для цього натискання завершено.

Якщо ліва кнопка миші не натиснута (`leftButtonState` дорівнює `HIGH`), змінна `leftButtonPressTime` скидається до нуля, що означає, що натискання кнопки завершено або не розпочиналося.

Аналогічний процес виконується для правої кнопки миші. Якщо права кнопка натиснута (`rightButtonState` дорівнює `LOW`), і `rightButtonPressTime` дорівнює нулю, зберігається поточний час натискання. Якщо натискання триває більше заданої тривалості, викликається функція `activateHotkey("right")`, яка активує гарячу клавішу для правої кнопки. Після активації гарячої клавіші змінна `rightButtonPressTime` скидається до нуля. Якщо права кнопка миші не натиснута (`rightButtonState` дорівнює `HIGH`), змінна `rightButtonPressTime` скидається до нуля.

На завершення додається невелика затримка (`delay(10)`) для стабільності роботи системи. Це запобігає надмірній кількості перевірок станів кнопок і забезпечує коректну обробку натискань.

Таким чином, функція `loop()` забезпечує безперервне відстеження стану кнопок миші і активує відповідні гарячі клавіші, якщо кнопки натиснуті протягом заданого часу. Це дозволяє користувачеві використовувати довгі натискання для виконання певних команд у програмному забезпеченні, наприклад, 3ds Max.

### **3.5.1 Налаштування гарячих кнопок у 3ds Max**

1. Відкрийте 3ds Max і перейдіть до меню "Customize" (Налаштування);
2. Виберіть "Customize User Interface" (Налаштування інтерфейсу користувача);
3. перейдіть на вкладку "Keyboard" (Клавіатура);
4. У полі "Category" (Категорія) виберіть потрібну категорію команд;
5. Виберіть команду, яку потрібно призначити гарячій клавіші;
6. У полі "Hotkey" (Гаряча клавіша) натисніть потрібну комбінацію клавіш (наприклад, Ctrl+S або Alt+F4, в даному випадку затискання лівої або правої кнопки периферійного пристрою) ;
7. Натисніть "Assign" (Призначити) і збережіть зміни

## **3.6 Методи покращення периферійного пристрою для системи автоматизованого проєктування**

### **3.6.1 Використання сенсора Pixart PMW3389**

Одним з найважливіших аспектів нашого пристрою є точність сенсора, який зчитує рухи користувача. Хоча сенсор MPU6050 забезпечує базову точність, його характеристики можна покращити за допомогою більш сучасних і точних сенсорів. Наприклад, для значного покращення точності та продуктивності нашого пристрою можна розглянути використання сенсора Pixart PMW3389. Це один з найсучасніших оптичних сенсорів, який використовується у високоякісних геймерських мишах. Він забезпечує надзвичайно високу точність і швидкість відстеження рухів, що може бути корисним для САПР.

Використання такого сенсора дозволить забезпечити надзвичайно точне і швидке відстеження рухів користувача, що є критично важливим для роботи у САПР.

Інтеграція сенсора Pixart PMW3389 у даний периферійний пристрій потребує врахування кількох ключових аспектів, таких як апаратне підключення, програмне забезпечення для обробки сигналу та налаштування сенсора.

Сенсор Pixart PMW3389 підключається до мікроконтролера через SPI інтерфейс, що забезпечує високу швидкість передачі даних. Типова схема підключення включає наступні сигнали: MOSI (Master Out Slave In), MISO (Master In Slave Out), SCK (Serial Clock) і CS (Chip Select).

```
// Приклад ініціалізації SPI інтерфейсу для роботи з Pixart PMW3389
#include <SPI.h>

const int CS_PIN = 10; // Пін для Chip Select

void setup() {
  pinMode(CS_PIN, OUTPUT);
  digitalWrite(CS_PIN, HIGH);
  SPI.begin();
}
```

а)

```
SPI.setClockDivider(SPI_CLOCK_DIV8); // Налаштування швидкості SPI
SPI.setDataMode(SPI_MODE3); // Налаштування режиму SPI
}

void loop() {
  // Ваш код для взаємодії з сенсором Pixart PMW3389
}
```

б)

Рисунок 3.13 – Ініціалізації SPI інтерфейсу для роботи з Pixart PMW3389 (а, б)

Для обробки сигналу від сенсора необхідно написати відповідні функції, які будуть зчитувати дані з сенсора через SPI інтерфейс. Наприклад, функція для зчитування поточних координат може виглядати наступним чином:



```
int readSensorData(byte reg) {  
    digitalWrite(CS_PIN, LOW);  
    SPI.transfer(reg); // Надсилаємо адресу регістра  
    int data = SPI.transfer(0x00); // Зчитуємо дані  
    digitalWrite(CS_PIN, HIGH);  
    return data;  
}
```

а)

```
void loop() {  
    int x = readSensorData(0x02); // Зчитуємо координату X  
    int y = readSensorData(0x03); // Зчитуємо координату Y  
    // Обробляємо зчитані дані  
}
```

б)

Рисунок 3.14 – Зчитування даних з сенсора через SPI інтерфейс (а, б)

Pixart PMW3389 має різні режими і налаштування, які можна змінювати для оптимізації його роботи під конкретні задачі.

```
void configureSensor() {  
    writeSensorData(0x0F, 0x01); // Встановлюємо роздільну здатність 800 DPI  
    writeSensorData(0x10, 0x00); // Встановлюємо режим безперервного зчитування  
}
```

а)

```
void writeSensorData(byte reg, byte value) {  
    digitalWrite(CS_PIN, LOW);  
    SPI.transfer(reg);  
    SPI.transfer(value);  
    digitalWrite(CS_PIN, HIGH);  
}
```

б)

Рисунок 3.15 – Налаштування параметрів (а, б)

Наприклад, можна налаштувати роздільну здатність (DPI) та інші параметри для досягнення оптимальної точності і швидкості відстеження.

### 3.6.2 Оптимізація програмного забезпечення

Програмне забезпечення нашого пристрою також може бути оптимізоване для підвищення продуктивності та точності. Одним з підходів є

покращення алгоритмів обробки даних з сенсора. Наприклад, можна застосувати більш складні методи фільтрації, такі як фільтр Калмана.

```
// Приклад використання фільтру Калмана для покращення точності даних сенсора
float kalmanFilter(float angle, float rate, float dt) {
    static float Q_angle = 0.001;
    static float Q_bias = 0.003;
    static float R_measure = 0.03;
    static float angle_err, bias = 0, rate_err;
    static float P[2][2] = {{0, 0}, {0, 0}};
```

a)

```
rate -= bias;
angle += dt * rate;

P[0][0] += dt * (dt * P[1][1] - P[0][1] - P[1][0] + Q_angle);
P[0][1] -= dt * P[1][1];
P[1][0] -= dt * P[1][1];
P[1][1] += Q_bias * dt;
```

б)

```
float S = P[0][0] + R_measure;
float K[2];
K[0] = P[0][0] / S;
K[1] = P[1][0] / S;

float y = angle - angle_err;
angle_err += K[0] * y;
bias += K[1] * y;
```

в)

```
float P00_temp = P[0][0];  
float P01_temp = P[0][1];  
  
P[0][0] -= K[0] * P00_temp;  
P[0][1] -= K[0] * P01_temp;  
P[1][0] -= K[1] * P00_temp;  
P[1][1] -= K[1] * P01_temp;  
  
return angle_err;
```

г)

Рисунок 3.16 – Приклад використання фільтра Калмана

Фільтр Калмана дозволяє прогнозувати наступні значення на основі попередніх даних, що значно підвищує точність вимірювань у динамічних умовах.

### 3.7 Інтеграція пристрою в інших галузях

Периферійний пристрій у вигляді бездротового пера для керування мишею, розроблений для систем автоматизованого проектування (САПР), має потенціал для широкого застосування в різних галузях. Завдяки своїм характеристикам, таким як висока точність, бездротова функціональність і зручність використання, цей пристрій може бути адаптований для використання в інших сферах, де важливі точність і ефективність керування

Даний периферійний пристрій може використовуватися в різних галузях. Наприклад, в галузі мистецтва, арт-дизайну, дистанційної освіти чи для домашнього використання.

Розберемо одну із галузей більш детально. Галузь мистецтва та арт-дизайну вимагає високої точності та зручності у використанні інструментів для створення візуальних творів. Бездротове перо, оснащене сенсором MPU6050, може стати незамінним інструментом для художників і арт-дизайнерів завдяки кільком ключовим перевагам.

По-перше, бездротове перо забезпечує високу точність і чутливість до рухів, що дозволяє художникам створювати детальні і точні зображення. Сенсор MPU6050, який відстежує рухи в шести ступенях свободи, дозволяє точно передавати найменші рухи руки, що є критичним для точного малювання та редагування графічних елементів.

По-друге, бездротова функціональність надає свободу руху, що є важливим для художників, які звикли працювати на великих полотнах або у просторих студіях. Художник може вільно пересуватися і працювати з різними частинами зображення, не обмежуючись довжиною кабелю.

Нарешті, зручність використання пристрою дозволяє художникам легко інтегрувати його в свої робочі процеси. Інтуїтивно зрозумілий інтерфейс і можливість налаштування різних функцій роблять його ідеальним інструментом для створення цифрових творів мистецтва.

### **3.8 Висновок до розділу 3**

У даному розділі було детально розглянуто середовище розробки для створення програмного забезпечення для апаратної частини. Використання Arduino IDE забезпечило простоту налаштування та програмування, що дозволило швидко розробляти і відлагоджувати код для мікроконтролерів. Інтеграція бібліотек і функцій для роботи з сенсорами та модулями Bluetooth значно спростила процес розробки.

Детально ознайомились із налаштуванням Bluetooth модулів HC-05 в режимах Master та Slave за допомогою AT команд. Це дозволило забезпечити надійне і стабільне з'єднання між пристроями, а також гнучко налаштовувати параметри модулів для безперебійної передачі даних.

Розробили User Story та Use Case діаграми для проекту, що допомогло візуально представити взаємодію користувача з пристроєм. Це сприяло кращому розумінню потреб користувачів та функціональних можливостей пристрою, що, в свою чергу, забезпечило ефективніше планування та розробку програмного забезпечення.

Створено програмне забезпечення для апаратної частини, що включало детальний опис коду для передавача та приймача. Було розглянуто процеси ініціалізації компонентів, калібрування сенсора, зчитування даних та їх обробку, а також передачу даних через Bluetooth.

Проаналізовано методи покращення периферійного пристрою для систем автоматизованого проектування. Було запропоновано використання точніших сенсорів, таких як Pixart PMW3389, оптимізацію програмного забезпечення, підвищення енергоефективності та покращення інтерфейсу користувача для підвищення точності, продуктивності та зручності використання пристрою.

Розглянуто інтеграцію пристрою в інших галузях, таких як мистецтво, арт-дизайн, дистанційна освіта та домашнє використання. Було виявлено, що пристрій може бути корисним для художників, дизайнерів, викладачів та домашніх користувачів, пропонуючи високоточне, зручне і надійне керування в різних сферах.

## ВИСНОВКИ

Згідно мети бакалаврської роботи розроблено периферійний пристрій для САПР 3Ds Max, який забезпечує покращену ергономіку, точність та ефективність роботи. У ході роботи було проаналізовано проблематику використання традиційних комп'ютерних мишей у контексті САПР, включаючи ергономічні проблеми, можливість виникнення СЗК та вплив цих аспектів на продуктивність. Розглянуто альтернативні рішення.

Було проаналізовано та обрано необхідні компоненти для створення пристрою, включаючи мікроконтролери, сенсори руху та бездротові модулі.

В ході розробки детально розглянуто всі етапи підключення та налаштування апаратної частини, включаючи сенсор MPU6050, модулі HC-05 та кнопки для емуляції кліків миші. Налаштовано бездротове з'єднання між модулями HC-05, що забезпечує передачу оброблених даних про рух та команди для емуляції миші. Було забезпечено стабільний канал зв'язку між мікроконтролерами за допомогою налаштування UART інтерфейсу.

Розроблено програмне забезпечення для апаратної частини, що включало детальний опис коду для передавача та приймача, алгоритми ініціалізації компонентів, зчитування та обробки даних з сенсора, передачу команд через Bluetooth та обробку натискань кнопок.

Також було розглянуто методи покращення пристрою, включаючи використання точніших сенсорів, оптимізацію програмного забезпечення, підвищення енергоефективності та покращення інтерфейсу користувача. Аналіз показав, що точніші сенсори, такі як Pixart PMW3389, можуть значно підвищити продуктивність та точність пристрою. Було досліджено можливість інтеграції пристрою в інших галузях. Була розроблена спеціальна частина з охорони праці.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Berezutsky V. I. Computer mouse, keyboard and carpal tunnel syndrome. *Medicni perspektivi (Medical perspectives)*. 2018. Vol. 23, no. 3(part1). P. 23–33. URL: [https://doi.org/10.26641/2307-0404.2018.3\(part1\).142330](https://doi.org/10.26641/2307-0404.2018.3(part1).142330) (date of access: 06.06.2024).
2. Hedge A., Feathers D., Rollings K. Ergonomic Comparison of Slanted and Vertical Computer Mouse Designs. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. 2010. Vol. 54, no. 6. P. 561–565. URL: <https://doi.org/10.1177/154193121005400604> (date of access: 06.06.2024).
3. Графічний планшет Huion H640P. URL: [https://rozetka.com.ua/ua/huion\\_h640p/p27885953/](https://rozetka.com.ua/ua/huion_h640p/p27885953/) (дата звернення: 30.04.2024).
4. Миша бездротова Logitech Lift Vertical Ergonomic Bluetooth Black (910-006495). URL: <https://hard.rozetka.com.ua/ua/logitech-910-006495/p432753758/> (дата звернення: 30.04.2024).
5. Arduino Micro та Arduino Pro Micro. URL: <https://all--arduino-ru.translate.goog/product/arduino-micro-i-arduino-pro-micro> (дата звернення: 10.04.2024).
6. PMW3389 Motion Sensor. URL: <https://www.tindie.com/products/citizenjoe/pmw3389-motion-sensor/> (дата звернення: 27.03.2024).
7. Акселерометр і гіроскоп MPU-6050. URL: <https://www.robostore.com.ua/ua/moduli-i-datchiki/sensory-datchiki-i-moduli/akselerometr-i-giroskop-mpu-6050/> (дата звернення: 10.04.2024).
8. Bruno R., Conti M., Gregori E. *Cluster Computing*. 2002. Vol. 5, no. 2. P. 117–131. URL: <https://doi.org/10.1023/a:1013989524865> (date of access: 06.06.2024).
9. I2C vs SPI. URL: <https://www.prodigytechno.com/i2c-vs-spi> (дата звернення: 10.04.2024).
10. Ryzhkov L., Zorin S. Complementary filter for determining the orientation of moving objects. *MECHANICS OF GYROSCOPIC SYSTEMS*. 2022. 2024 р.

No. 42. P. 63–68. URL: <https://doi.org/10.20535/0203-3771422021268465> (date of access: 06.06.2024).

11. AT-команди Bluetooth HC-05. URL: <https://wiki-arduino-ru.translate.google.com/page/at-komandy-bluetooth-hc-05> (дата звернення: 10.04.2024).



## ДОДАТОК А

**Довідка**  
**про перевірку на унікальність пояснювальної записки**  
**бакалаврської кваліфікаційної роботи на тему:**  
**«Периферійний пристрій для систем автоматизованого проектування»**

студента спеціальності 123 «Комп'ютерна інженерія», 405 групи  
Горішна Ольга Михайлівна  
прізвище, ім'я, по-батькові

Перевірку тексту здійснено сервісом: онлайн-сервіс Unicheck

Результат перевірки тексту кваліфікаційної бакалаврської роботи: схожість складає 2,82 %.

The screenshot shows the Unicheck report for user Ivan Burlyachenko. It includes details such as the check date (08.06.2024 19:58:31 EEST), report date (08.06.2024 20:08:42 EEST), and file information (file name: Горішна\_О.М\_Комп'ютерна\_інженерія\_КБР\_2024, page count: 52, word count: 11637, character count: 90494, file size: 101.49 KB, file ID: 1016136797). The report highlights a 2.82% match rate, with 2.7% from internet sources (324 matches) and 1.01% from library sources (135 matches). It also shows 0% quotes and 0% exclusions. A Modifind section indicates 21 replaced characters.

Здобувач:

студентка 405 групи

\_\_\_\_\_ О. М. Горішна  
підпис ініціали, прізвище

Дата: « \_\_\_\_ » \_\_\_\_\_ 2024 р.

Керівник:

ст. викладач

\_\_\_\_\_ І. С. Бурлаченко  
підпис ініціали, прізвище

## ДОДАТОК Б

### Код для налаштування Bluetooth модулів HC-05 в режимах Master та Slave за допомогою AT команд

```
//Початок скетчу

#include <SoftwareSerial.h>

//          ім'я          пароль          швидкість
//   CMODE      отримати адресу      записати адресу      роль
//запит      AT+NAME?      AT+PSWD?      AT+UART?
//   AT+CMODE?      AT+ADDR?      AT+BIND?      AT+ROLE?

//варіант
відповіді +NAME:SLAVE      +PIN:"1234"      +UART:9600,0,0      +CMOD
E:1      AT+BIND=98d3,71,feaefff      +ROLE:0

//-----
//

//для слейва |
AT+NAME=SLAVE      AT+PSWD="1234"      AT+UART=9600,0,0      AT+CMODE=1      +AD
DR:98d3:71:feaefff      AT+ROLE=0

//для мастера |
AT+NAME=MASTER      AT+PSWD="1234"      AT+UART=9600,0,0      AT+CMODE=0
      AT+BIND=98d3,71,feaefff      AT+ROLE=1

//скидання налаштувань AT+ORGL це для крайнього випадку, якщо щось наплутали
з налаштуваннями, і не можете знайти як виправити

#include <SoftwareSerial.h>

// используемые пины
const int arduino_rx = 5;
const int arduino_tx = 6;
SoftwareSerial mySerial (arduino_rx, arduino_tx);

void setup() {
    pinMode(arduino_rx, INPUT);
    pinMode(arduino_tx, OUTPUT);
}
```

```
Serial.begin(9600); //в моніторі порта вибрати NL&CR, швидкість Arduino з
комп'ютером 9600

mySerial.begin(38400); // по замовчуванню швидкість Bluetooth модуля с
Arduino в режимі AT команд 38400 іноді 9600

Serial.println("<<< Start! >>>");
mySerial.println("AT");
}

void loop() {
  if (mySerial.available()) {
    char c = mySerial.read(); // читаємо з програмного Serial-порта
    Serial.print(c); // пишемо в штатний Serial-порт
  }
  if (Serial.available()) {
    char c = Serial.read(); // читаємо зі штатного Serial-порта
    mySerial.write(c); // пишемо в програмний Serial-порт
  }
}
//кінець скетчу
```

## ДОДАТОК В

### Код передавача пристрою

```
#include <Wire.h>
#include <I2Cdev.h>
#include <MPU6050.h>

MPU6050 mpu;

float gyroXOffset = 0;
float gyroYOffset = 0;

const float alpha = 0.2;
const float baseSensitivity = 0.1;
const float speedFactor = 2.0;

const int leftButtonPin = 4; // Пін для лівої кнопки миші
const int rightButtonPin = 5; // Пін для правої кнопки миші

void setup() {
  Serial.begin(9600); // Ініціалізація серійного зв'язку для відлагодження
  Serial1.begin(9600); // Ініціалізація апаратного серійного зв'язку для
Bluetooth

  Wire.begin();
  pinMode(leftButtonPin, INPUT_PULLUP);
  pinMode(rightButtonPin, INPUT_PULLUP);

  mpu.initialize();
  if (!mpu.testConnection()) {
    Serial.println("MPU6050 connection failed");
    while (1);
  }
}
```

```
    calibrateGyroscope();
    Serial.println("Setup complete");
}

void loop() {
    int16_t gx, gy, gz;
    mpu.getRotation(&gx, &gy, &gz); // Отримання даних з гіроскопа

    float gyroX = gx / 131.0 - gyroXOffset;
    float gyroY = gy / 131.0 - gyroYOffset;

    static float filteredX = 0;
    static float filteredY = 0;

    // Згладжування сигналу
    filteredX = alpha * gyroX + (1 - alpha) * filteredX;
    filteredY = alpha * gyroY + (1 - alpha) * filteredY;

    int leftButtonState = digitalRead(leftButtonPin);
    int rightButtonState = digitalRead(rightButtonPin);

    // Надсилання даних через блютуз
    Serial1.print(filteredX);
    Serial1.print(",");
    Serial1.print(filteredY);
    Serial1.print(",");
    Serial1.print(leftButtonState);
    Serial1.print(",");
    Serial1.println(rightButtonState);

    delay(5); // Невелика затримка для стабільності
}
```

```
void calibrateGyroscope() {  
    float sumX = 0, sumY = 0;  
    int numSamples = 100;  
  
    for (int i = 0; i < numSamples; i++) {  
        int16_t gx, gy, gz;  
        mpu.getRotation(&gx, &gy, &gz);  
        sumX += gx / 131.0;  
        sumY += gy / 131.0;  
        delay(10);  
    }  
  
    gyroXOffset = sumX / numSamples;  
    gyroYOffset = sumY / numSamples;  
  
    Serial.print("gyroXOffset: ");  
    Serial.println(gyroXOffset);  
    Serial.print("gyroYOffset: ");  
    Serial.println(gyroYOffset);  
}
```

## ДОДАТОК Г

### Код приймача пристрою

```
#include <Mouse.h>

const float baseSensitivity = 0.1; // Збільшено базову чутливість для швидшого руху
const float speedFactor = 2.0; // Збільшено фактор швидкості для більш швидкого руху

void setup() {
    Serial.begin(9600);
    Serial1.begin(9600); // Ініціалізація апаратного серійного зв'язку для Bluetooth
    Mouse.begin();
    Serial.println("Setup complete");
}

void loop() {
    if (Serial1.available()) {
        // Зчитування даних з Bluetooth
        String data = Serial1.readStringUntil('\n');
        Serial.println("Received data: " + data); // Відображення прийнятих даних

        // Розбір даних
        float gyroX = data.substring(0, data.indexOf(',')).toFloat();
        data = data.substring(data.indexOf(',') + 1);
        float gyroY = data.substring(0, data.indexOf(',')).toFloat();
        data = data.substring(data.indexOf(',') + 1);
        int leftButtonState = data.substring(0, data.indexOf(',')).toInt();
        int rightButtonState = data.substring(data.indexOf(',') + 1).toInt();

        Serial.print("gyroX: ");
        Serial.print(gyroX);
```

```
Serial.print(" gyroY: ");
Serial.print(gyroY);
Serial.print(" leftButtonState: ");
Serial.print(leftButtonState);
Serial.print(" rightButtonState: ");
Serial.println(rightButtonState);

// Обчислення руху миші
int moveX = baseSensitivity * speedFactor * gyroX;
int moveY = baseSensitivity * speedFactor * gyroY;

if (abs(moveX) < 1) moveX = 0;
if (abs(moveY) < 1) moveY = 0;

Serial.print("moveX: ");
Serial.print(moveX);
Serial.print(" moveY: ");
Serial.println(moveY);

// Рух миші
Mouse.move(moveX, moveY);

// Обробка натискань кнопок миші
if (leftButtonState == LOW) {
    Mouse.press(MOUSE_LEFT);
} else {
    Mouse.release(MOUSE_LEFT);
}

if (rightButtonState == LOW) {
    Mouse.press(MOUSE_RIGHT);
} else {
    Mouse.release(MOUSE_RIGHT);
}
```



```
}  
  
    delay(10); // Невелика затримка для стабільності  
}  
}
```

## ДОДАТОК Д

### Графічні матеріали кбр

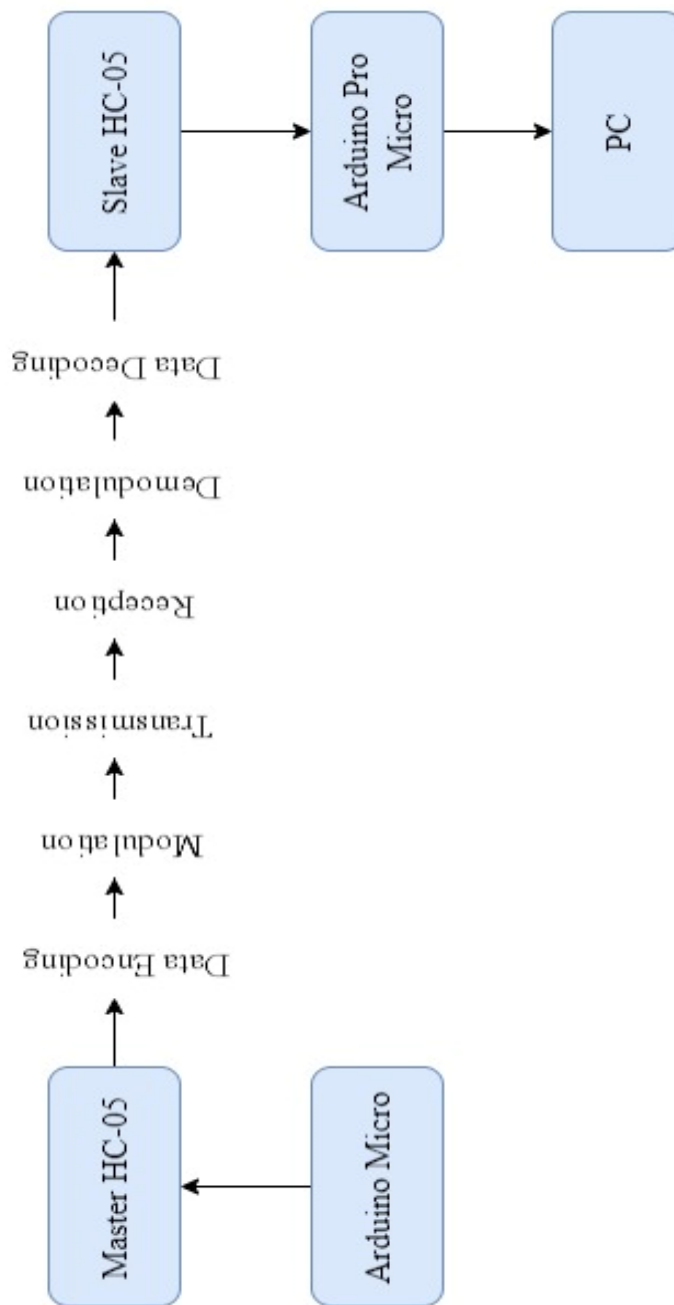


Рисунок Д.1 – Процес передачі даних між двома модулями HC-05

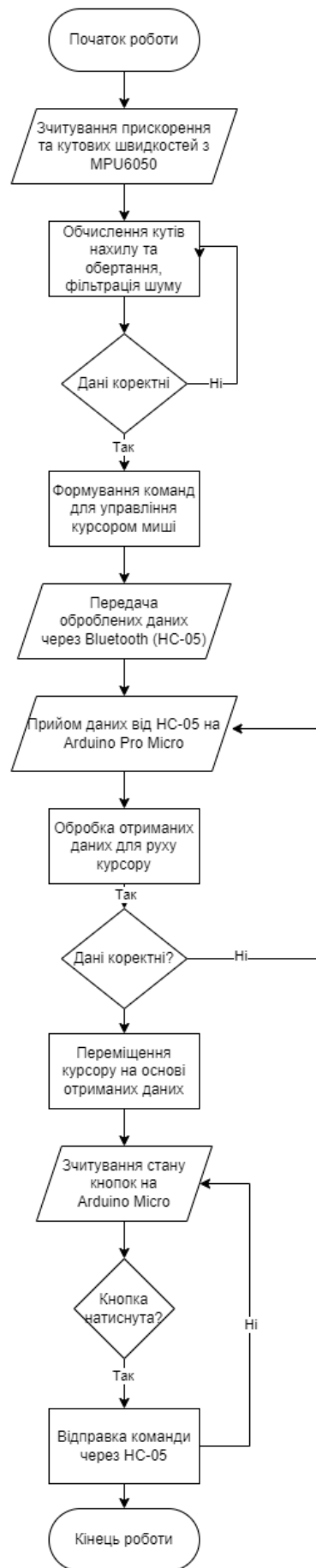


Рисунок Д.2 – Блок-схеми алгоритму роботи пристрою

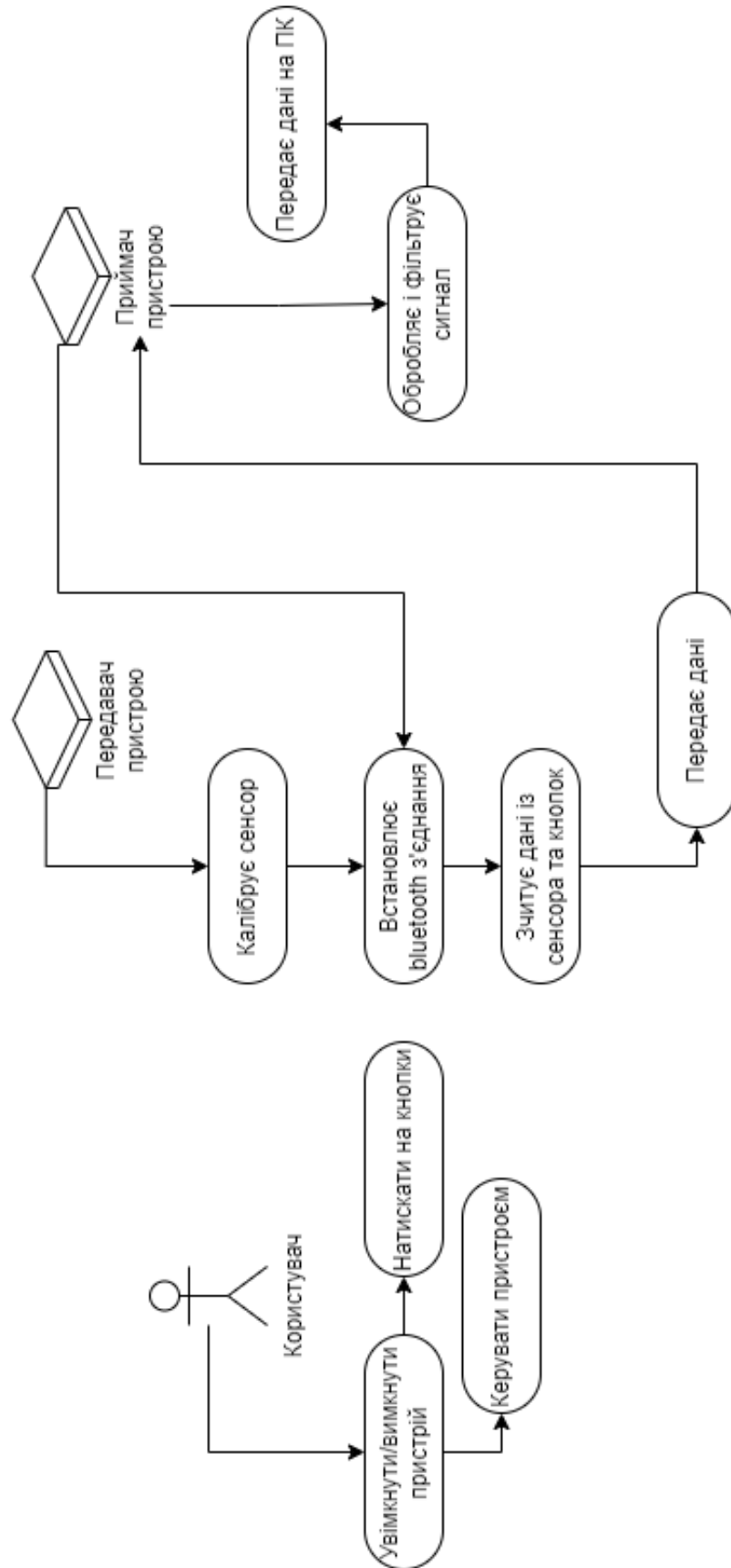


Рисунок Д.3 – Use Case діаграма пристрою