

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерної інженерії та управління _____
Кафедра _____ Автоматизації проєктування обчислювальної техніки _____
Рівень вищої освіти _____ перший (бакалаврський) _____
Спеціальність _____ 123 Комп'ютерна інженерія _____
Тип програми _____ Освітньо-професійна _____
Освітня програма _____ Комп'ютерна інженерія _____

ЗАТВЕРДЖУЮ:

Зав. _____ кафедри _____

(підпис)

« _____ » _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Косенко Святославу Олександровичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Система позиціонування в просторі за допомогою інерційного датчика _____

затверджена наказом університету від _21_ ____ _05_ ____ 2025 р. № 403Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії _10_ ____ _06_ ____ 2025 р.

3. Вихідні дані до роботи _____

Arduino

Програмно-апаратний модуль

4. Перелік питань, що потрібно опрацювати в роботі _____

Аналіз та огляд існуючих систем.

Постановка задачі.

Розробка структурної схеми пристрою.

Розробка функціональної схеми програми.

Розробка алгоритму роботи пристрою.

Тестування

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) _____
13 слайдів

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Видача теми проекту, узгодження і затвердження теми	06.05.2025 – 09.05.2025	
2	Аналіз проблемної галузі, постановка задачі, вибір інструментальних засобів	09.05.2025 – 14.05.2025	
3	Розробка структурної схеми пристрою, вибір апаратної платформи	14.05.2025 – 16.05.2025	
4	Розробка функціональної схеми програми	16.05.2025 – 17.05.2025	
5	Розробка програмних модулів. Проведення тестування	17.05.2025 – 23.05.2025	
6	Оформлення пояснювальної записки	23.05.2025 – 25.05.2025	
7	Перевірка виконаного проекту керівником, допуск до захисту	25.05.2025 – 10.06.2025	

Дата видачі завдання _____ 06.05.2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Філіппенко І.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи містить: 48 сторінок, 12 рисунків, 15 джерел посилання.

МОДУЛЬ, ПРОЄКТ, МІКРОКОНТРОЛЕР, MEMS

Метою кваліфікаційної роботи є розробка програмно-апаратного системи позиціонування в просторі за допомогою інерційного датчика на базі Arduino. Було проведено огляд існуючих програмно-апаратних модулів позиціонування, їхніх характеристик, переваг та недоліків. У ході виконання кваліфікаційної роботи було складено технічне завдання на розробку. За технічним завданням підібрано апаратну частину системи. Було запропонована структурна та функціональна модель модулю. Виконано вибір мікроконтролера, елементної бази та комутації. Проведено тестування пристрою.

ABSTRACT

The explanatory note of the qualification work contains: 48 pages, 12 pictures, 15 sources of links.

MODULE, PROJECT, MICROCONTROLLER, MEMS

The purpose of the qualification work is to develop a software and hardware system for positioning in space using an inertial sensor based on Arduino. A review of existing software and hardware positioning modules, their characteristics, advantages and disadvantages was conducted. During the qualification work, a technical task for development was drawn up. The hardware part of the system was selected according to the technical task. A structural and functional model of the module was proposed. The microcontroller, element base and switching were selected. The device was tested.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	7
ВСТУП.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	11
1.1 MEMS-сенсори для визначення положення тіла у просторі.....	11
1.2 MEMC-акселерометри	14
1.3 MEMC-гіроскопи.....	15
1.4 Порівняльна характеристика MEMS-сенсорів.....	17
2 РОЗРОБКА АПАРАТНОГО МОДУЛЯ	21
2.1 Структурна схема пристрою	21
2.2 Вибір мікроконтролера	21
2.3 MPU-6050.....	24
2.4 OLED-дисплей.....	27
3 РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ.....	30
4 ТЕСТУВАННЯ ПРИСТРОЮ	41
ВИСНОВКИ	46
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	47
ДОДАТОК А. Графічний матеріал кваліфікаційної роботи	49

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ

MEMS (Micro-Electro-Mechanical Systems) – мікроелектромеханічні системи

ATmega328P – мікроконтролер, який використовується в платі Arduino

GND – точка нульового потенціалу мікросхеми

IDE – інтегроване середовище розробки, програмне забезпечення для написання та компіляції програм

I/O – вхід/вихід, порти вводу/виводу мікроконтролера

МК – мікроконтролер

PWM – широтно-імпульсна модуляція, метод керування аналоговими сигналами за допомогою цифрових сигналів

UART – універсальний асинхронний приймач/передавач

USB – універсальна послідовна шина – інтерфейс передачі даних для підключення периферійних пристроїв

ВСТУП

Мікроелектромеханічні системи (MEMS, Micro-Electro-Mechanical Systems) це технології, що інтегрують мікроелектроніку з механічними компонентами, створюючи мініатюрні пристрої, які поєднують рухомі частини з електронними схемами для вирішення механічних задач, таких як вимірювання, переміщення чи керування, на мікроскопічному рівні. Ці системи вирізняються від інших компактними розмірами.

Вони можуть бути за розмірами як від кількох мікрометрів, так до кількох міліметрів, що дає змогу легко вбудовувати їх у різноманітні електронні пристрої без помітного збільшення габаритів. Також завдяки своїм розмірам від мікрометрів до міліметрів MEMS ідеально підходять для розміщення на підложці кристалу, дозволяючи об'єднувати механічні елементи, такі як сенсори чи приводи, з електронними схемами в єдині мініатюрні системи.

Завдяки високій точності та чутливості до найменших змін такі пристрої знаходять застосування там, де потрібна прецизійність, наприклад, у медичних технологіях чи навігаційних системах, а їхня здатність працювати з мінімальним енергоспоживанням робить їх ідеальним вибором для мобільних і портативних пристроїв.

Різнноманітні сенсори на основі MEMS-технологій стають основою для створення сучасних невеликих за розміром проектів. MEMS-сенсори знаходять широке застосування в різних сферах, таких як мобільні пристрої, медичні технології, спортивні тренування та робототехніка. Врахування факторів, таких як шум, дрейф та взаємодія між сенсорами, є ключовим для створення надійних і точних систем. Окрім того, розробка таких систем повинна передбачати інтеграцію MEMS-сенсорів у більші апаратні платформи з урахуванням енергоспоживання, компактності та зручності

використання, особливо в носимих пристроях, що активно використовуються в медицині та фітнес-технологіях.

Одними з найбільш поширених MEMS-сенсорів є акселерометри, гіроскопи та магнітометри, які дозволяють вимірювати лінійне прискорення, кутову швидкість та орієнтацію об'єкта. Ці сенсори взаємодіють між собою, забезпечуючи високу точність у визначенні положення тіла, що є критично важливим для таких систем, як навігація, моніторинг руху та реабілітація та інше.

Акселерометри та гіроскопи – це сенсори, які завдяки своїй здатності вимірювати прискорення та кутову швидкість, знайшли широке застосування в багатьох сферах: від смартфонів, де вони визначають орієнтацію екрана й відстежують рухи в іграх чи доповненій реальності, до автомобільної промисловості, де допомагають у системах стабілізації, подушках безпеки та навігації. В авіації й космічних технологіях для інерційних систем орієнтації там, де GPS недоступний. У робототехніці вони забезпечують рівновагу й точність рухів, у фітнес-пристроях рахують кроки й аналізують активність, а у віртуальній реальності та геймінгу створюють ефект занурення через відстеження положення контролерів чи голови. В промисловості вони використовуються для моніторингу вібрацій машин. В медицині ці пристрої знайшли застосування у протезах чи пристроях для виявлення падінь. MEMS-сенсори становляться незамінними в сучасних технологіях, особливо коли вони поєднуються з іншими системами для створення розумних і автономних рішень.

Особливу увагу слід приділити можливостям MEMS-технологій у медицині та спортивних дослідженнях, де точне вимірювання положення тіла є ключовим для реабілітації пацієнтів та покращення спортивних результатів. Застосування таких сенсорів у носимих пристроях дозволяє відстежувати рухи тіла та коригувати техніку вправ або стан здоров'я користувача.

У результаті, MEMS-системи стали основою для безлічі сучасних пристроїв, забезпечуючи нові можливості в області точного вимірювання, навігації, медичних технологій та багатьох інших сфер.

Дана кваліфікаційна робота направлена на вивчення принципів роботи з MEMS-сенсорами, а саме акселерометрами та гіроскопами, їх характеристиками, принципами роботи, їх застосування для вимірювання положення тіла в просторі та ключові аспекти розробки систем на їх основі.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 MEMS-сенсори для визначення положення тіла у просторі

Сенсори на основі MEMS (Micro-Electro-Mechanical Systems) займають важливе місце в сучасних технологіях для визначення положення тіла в просторі. Завдяки своїм мініатюрним розмірам, низькому енергоспоживанню та високій точності, MEMS-сенсори знайшли широке застосування в різних галузях, включаючи мобільні пристрої, медичні технології, автоматизацію та робототехніку.

Для визначення положення тіла у просторі зазвичай використовуються такі типи MEMS-сенсорів, як акселерометри, гіроскопи та магнітометри. Вони дозволяють отримувати дані про лінійне прискорення, кутову швидкість та орієнтацію об'єкта, що є необхідним для створення систем точної навігації та контролю руху. Комбінація цих сенсорів дозволяє не лише визначати нахил і позицію в реальному часі, але й коригувати погрішності в навігаційних системах за допомогою фільтрації сигналів та алгоритмів синхронізації. Використання MEMS-сенсорів для визначення положення тіла стало важливим кроком до розвитку інтелектуальних систем управління рухом, що застосовуються в широкому спектрі відомих і нових технологій..

Гіроскоп і акселерометр – два ключові сенсори, які широко застосовуються для вимірювання рухів та орієнтації, але кожен з них виконує своє унікальне завдання. Для визначення головних задач цих пристроїв для вирішення визначення положення тіла у просторі розглянемо основні функції.

Гіроскоп – це сенсор, який вимірює кутову швидкість обертання пристрою навколо осей. Він фіксує, з якою швидкістю та в якому напрямку відбувається обертання або нахил пристрою. Гіроскопи допомагають визначати точне положення пристрою у просторі, відстежуючи зміни його

орієнтації. Їх використання особливо актуальне в таких областях, як стабілізація камери (запобігання розмиттю зображення при русі), а також у навігації та системах контролю напрямку. Тобто здатність відслідковувати зміни напрямку і нахилу гіроскопа дозволяє застосовувати його для стабілізації зображення в камерах або у навігаційних системах.

Акселерометр, в свою чергу, вимірює лінійне прискорення по одній або кільком осям, включаючи гравітаційне. Він фіксує зміни прискорення, що впливають на пристрій, а також визначає його положення (наприклад, вертикальне або горизонтальне) та нахил. Акселерометри широко використовуються для розпізнавання рухів пристрою, наприклад коли смартфон виявляє поворот екрана або коли фітнес-трекер відстежує фізичну активність.

Акселерометри та гіроскопи часто використовуються в парі для досягнення більш точної та надійної оцінки положення та руху об'єкта в просторі. Коли акселерометр фіксує нахил пристрою, гіроскоп відстежує його обертання. У комбінації вони стають невід'ємною частиною різних технологій, включаючи смартфони, геймпади, квадрокоптери та інші пристрої, які потребують точного відстеження рухів та орієнтації. Кожен з цих сенсорів має свої переваги та обмеження, тому їх комбінація дозволяє створити системи, які ефективно компенсують недоліки кожного з них, підвищуючи загальну точність вимірювань.

Комбінація акселерометра та гіроскопа необхідна для точного визначення орієнтації. А саме, акселерометр вимірює лінійне прискорення і допомагає визначити нахил або орієнтацію пристрою відносно земної поверхні, особливо в умовах постійного гравітаційного поля. Водночас, гіроскоп відслідковує зміну кута повороту (углову швидкість) та дозволяє точно визначити, як об'єкт змінює своє положення в просторі. Коли ці два сенсори працюють разом, система може постійно коригувати дані, компенсуючи похибки акселерометра за рахунок високої чутливості

гіроскопа, особливо при швидких рухах або при відсутності чіткої гравітаційної орієнтації.

У багатьох пристроях, таких як смартфони, безпілотники та спортивні трекери, комбінація акселерометра та гіроскопа використовується для надання точних даних про рух. Акселерометр може виявити прискорення і напрямок, а гіроскоп – кутову швидкість, що дозволяє точніше відслідковувати складні рухи, наприклад, повороти або зміни швидкості.

Використання обох сенсорів у стабілізаційних системах, таких як стабілізатори камери або автопілоти в літаках та дронів, дозволяє підтримувати стабільне положення навіть при різких змінах руху. Акселерометри допомагають визначити зміщення відносно горизонтальної осі, а гіроскопи – фіксувати обертання, що дає змогу компенсувати ці зміни та зберігати стабільність.

Для моніторингу фізичної активності, таких як вимірювання кроків, аналіз рухів або пози, комбінація акселерометра і гіроскопа дозволяє більш точно визначати не лише кількість виконаних рухів, а й їх інтенсивність, швидкість та напрямок. Це корисно, наприклад, у спортивних трекерах або медичних пристроях для реабілітації, що відслідковують положення тіла пацієнта.

У смартфонах та ігрових консолях ці сенсори дозволяють відслідковувати рухи користувача, що дає змогу управління через жести або зміну орієнтації екрану. Віртуальні реальності (VR) та доповнена реальність (AR) також використовують цю комбінацію для забезпечення більш реалістичних і точних відгуків на рухи користувача.

Тому можна визначити наступні переваги використання обох сенсорів разом:

- підвищення точності вимірювань завдяки використанню різних типів сенсорів для різних аспектів руху;
- забезпечення більш стабільних даних в умовах шуму чи вібрацій, оскільки кожен сенсор може компенсувати похибки іншого;

– можливість використовувати у складних системах, що потребують точного визначення положення та руху в реальному часі.

1.2 МЕМС-акселерометри

Принцип дії сенсорів руху (акселерометрів і гіроскопів) полягає у вимірюванні зсуву інерційної маси відносно корпусу пристрою та перетворенні цього зсуву на пропорційний електричний сигнал. Тобто в таких МЕМС-пристроях використовують камертонну систему електродів: дві підвішені маси коливаються в протилежних напрямках.

Конструкція ємнісного акселерометра базується на використанні декількох пластин: частина з них закріплена стаціонарно, а інші можуть вільно переміщатися в межах корпусу. Ці пластини утворюють конденсатор, ємність якого змінюється залежно від відстані між ними. Конденсатори підключені до кола резонансного генератора. Схема такого акселерометра та принцип дії наведено на рисунку 1.1.

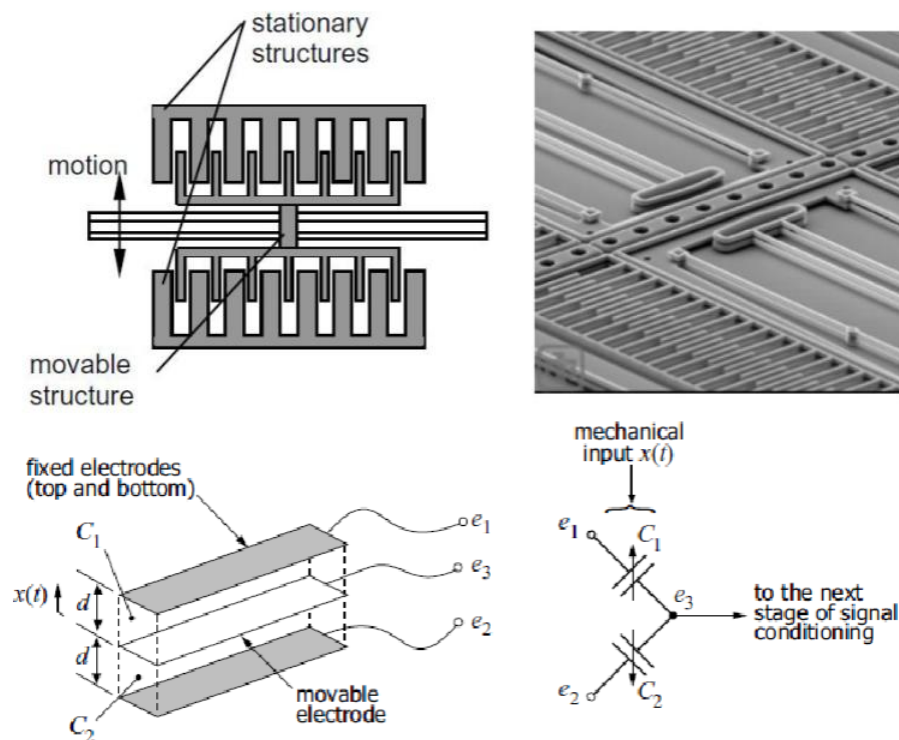


Рисунок 1.1 – МЕМС-акселерометр

Процес роботи можна описати таким чином: прискорення викликає механічне зміщення маси, це зміщення змінює ємність, а електроніка обробляє ці зміни, видаючи цифровий або аналоговий сигнал, який відповідає величині та напрямку прискорення.

Ця зміна ємності фіксується електронною схемою, яка також розміщена на тій самій підложці, і перетворюється в електричний сигнал (наприклад, напругу), пропорційний величині прискорення. Для тривимірного вимірювання використовуються три такі структури, розташовані перпендикулярно одна до одної, щоб охопити осі X, Y і Z.

Їхня чутливість дозволяє виявляти навіть найменші рухи чи нахили, хоча точність може залежати від зовнішніх факторів, таких як температура чи вібраційний шум. Завдяки мікроскопічним розмірам (від кількох мікрометрів до міліметрів), низькому енергоспоживанню та можливості інтеграції з мікросхемами, MEMS-акселерометри стали основою для багатьох пристроїв, від смартфонів до автомобільних систем.

1.3 MEMS-гіроскопи

Принцип роботи MEMS-гіроскопа ґрунтується на вимірюванні кутової швидкості, тобто швидкості обертання об'єкта навколо певної осі, і найчастіше базується на ефекті Коріоліса. Основним елементом такого гіроскопа є мікроскопічна механічна структура, яка складається з рухомої маси (або кількох мас), підвішеної на пружних елементах і закріпленої на підложці кристалу. Ця маса приводиться в коливальний рух за допомогою електростатичних або п'єзоелектричних приводів, які змушують її вібрувати в заданому напрямку з постійною частотою — це називається первинним рухом.

Коли об'єкт, на якому встановлено гіроскоп, починає обертатися, виникає сила Коріоліса, перпендикулярна як до напрямку обертання, так і до первинного коливання маси. Ця сила викликає вторинне зміщення маси в

іншому напрямку, пропорційне кутовій швидкості обертання. У більшості MEMS-гіроскопів це зміщення вимірюється ємнісним методом: рухома маса утворює конденсатор із нерухомими електродами, і її зміщення під впливом сили Коріоліса змінює відстань між пластинами, а отже, і ємність. Електронна схема, інтегрована на тій самій підложці, фіксує ці зміни, перетворюючи їх в електричний сигнал (аналоговий або цифровий), який відображає величину та напрямок кутової швидкості.

Для тривимірного вимірювання MEMS-гіроскопи мають три такі структури, орієнтовані вздовж осей X, Y і Z. Завдяки мікроскопічним розмірам (від кількох мікрометрів до міліметрів) і можливості вбудовування в мікросхеми, вони широко застосовуються в смартфонах, дронах, ігрових контролерах і навігаційних системах. Однак їхня точність може знижуватися через дрейф показів із часом або вплив температури, що вимагає калібрування чи комбінування з іншими сенсорами, наприклад, акселерометрами, для підвищення надійності даних.

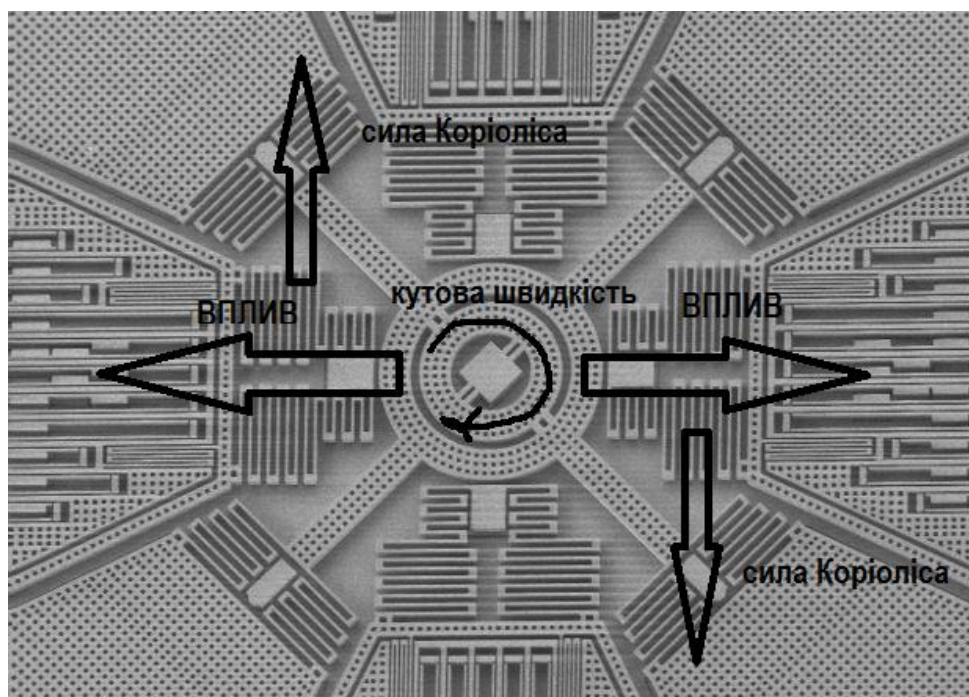


Рисунок 1.2 – Принципова діаграма MEMS-гіроскопа

1.4 Порівняльна характеристика MEMS-сенсорів

Проведемо порівняльну характеристику самих поширених MEMS-сенсорів для визначення положення тіла у просторі за їх основними параметрами. Оберемо пристрої відомих провідних виробників, таких як STMicroelectronics, Bosch Sensortec, Analog Devices та InvenSense.

Акселерометр LIS3DH фірми STMicroelectronics має наступні параметри:

- 3-осьовий акселерометр;
- діапазон вимірювання $\pm 2g$, $\pm 4g$, $\pm 8g$, $\pm 16g$ (налаштовується);
- чутливість до 12 біт (залежить від діапазону, наприклад, 1 мг/LSB при $\pm 2g$);
- енергоспоживання 11 мкА в нормальному режимі, 2 мкА в режимі низького споживання»
- I2C, SPI.

LIS3DH має вбудований буфер FIFO на 32 рівні, також має функцію самотестування, компактний корпус (3x3x1 мм LGA). Може використовуватись в розробках таких, як смартфони, фітнес-трекери, промислові датчики.

LSM6DSL фірми STMicroelectronics – це комбінований пристрій акселерометр та гіроскоп. Має наступні параметри:

- 6-осьовий інерційний модуль, а саме 3 осі акселерометра, та 3 осі гіроскопа;
- діапазон вимірювання акселерометра $\pm 2g$, $\pm 4g$, $\pm 8g$, $\pm 16g$ в залежності від налаштування;
- діапазон вимірювання гіроскопа ± 125 , ± 250 , ± 500 , ± 1000 , ± 2000 dps (градусів за секунду) в залежності від налаштування;
- чутливість акселерометра до 0.061 мг/LSB ($\pm 2g$);
- чутливість гіроскопа до 4.375 мдps/LSB (± 125 dps);

– енергоспоживання складає 0.65 мА в комбінованому режимі, 0.4 мА – якщо використовується тільки акселерометр.

– I2C, SPI.

LSM6DSL має вбудований процесор для обробки даних, FIFO на 8 кБ, також має високу стійкість до ударів (до 20 000g).

Може використовуватись в розробках таких, як дрони, VR-системи, робототехніка.

MPU-6050 фірми InvenSense/TDK це також комбінований пристрій акселерометр, гіроскоп та датчик температури. Має наступні параметри:

- 6-осьовий інерційний модуль;
- діапазон вимірювання акселерометра $\pm 2g, \pm 4g, \pm 8g, \pm 16g$;
- діапазон вимірювання гіроскопа $\pm 250, \pm 500, \pm 1000, \pm 2000$ dps;
- чутливість акселерометра до 16384 LSB/g ($\pm 2g$);
- чутливість гіроскопа до 131 LSB/dps (± 250 dps);
- енергоспоживання складає 3.9 мА в повному режимі, 10 мкА в режимі сну;

– I2C.

MPU-6050 має вбудований цифровий процесор руху (DMP), популярний у DIY-проектах та має невелику ціна.

Може використовуватися в таких розробках як стабілізація камер, прості дрони.

BNO055 фірми Bosch Sensortec це також комбінований модуль, в склад якого входить акселерометр, гіроскоп та магнітометр.

– 9-осьовий інерційний модуль, а саме 3 осі акселерометра, 3 осі гіроскопа, 3 осі магнітометра;

- діапазон вимірювання акселерометра $\pm 2g, \pm 4g, \pm 8g, \pm 16g$;
- діапазон вимірювання гіроскопа $\pm 125, \pm 250, \pm 500, \pm 1000, \pm 2000$ dps;
- чутливість акселерометра до 1 мг/LSB ($\pm 2g$);
- чутливість гіроскопа до 0.03 dps/LSB (± 125 dps);

– енергоспоживання складає 12 мА в повному режимі, значно нижче в окремих режимах.

– I2C, UART.

BNO055 має вбудований процесор для об'єднання даних (sensor fusion), видає кватерніони та кути Ейлера, має високу точність.

Може використовуватися для навігації, робототехніки, професійних дронів.

ADXL345 фірми Analog Devices – це акселерометр, який має наступні параметри:

– 3-осьовий акселерометр;

– діапазон вимірювання акселерометра $\pm 2g, \pm 4g, \pm 8g, \pm 16g$;

– чутливість акселерометра до 3.9 мг/LSB ($\pm 2g, 10$ біт), до 13 біт у високій роздільності;

– енергоспоживання складає 40 мкА в режимі вимірювання, 0.1 мкА в режимі очікування.

– I2C, SPI.

BNO055 має високу роздільність, функцію виявлення падіння та ударів, компактний корпус (3x5x1 мм LGA).

Може використовуватися в промислових системах, медичних пристроях, геймінгу.

З порівняння характеристик мікросхем можна побачити наступне:

1) MPU-6050 є найдешевшим і найпопулярнішим, тоді як BNO055 дорожчий через додатковий магнітометр і складну обробку даних;

2) BNO055 і LSM6DSL виграють завдяки вбудованим процесорам і вищій чутливості, тоді як MPU-6050 може мати більший дрейф у гіроскопі;

3) за енергоспоживанням LIS3DH і ADXL345 лідирують для застосувань із низьким споживанням, а MPU-6050 і BNO055 більше підходять для активних систем;

4) за функціональністю можна визначити, що BNO055 є найкращий для навігації завдяки 9 осям, LSM6DSL – більш зручний для динамічних

застосувань (дрони, VR), а LIS3DH і ADXL345 – можна використовувати для простих статичних вимірювань.

Таблиця 1.1 – Порівняння характеристик

Мікросхема	Тип	Діапазон (accel / gyro)	Чутливість (accel / gyro)	Енергоспо- живання	Інтерфейс
LIS3DH	акселерометр	$\pm 2g$ – $\pm 16g$	1 мг/LSB	11 мкА	I2C, SPI
LSM6DSL	акселерометр/ гіроскоп	$\pm 2g$ – $\pm 16g$ / ± 125 – ± 2000 dps	0.061 мг/LSB / 4.375 мдps	0.65 мА	I2C, SPI
MPU-6050	акселерометр/ гіроскоп	$\pm 2g$ – $\pm 16g$ / ± 250 – ± 2000 dps	16384 LSB/g / 131 LSB/dps	3.9 мА	I2C
BNO055	акселерометр/ гіроскоп/ магнітометр	$\pm 2g$ – $\pm 16g$ / ± 125 – ± 2000 dps	1 мг/LSB / 0.03 dps/LSB	12 мА	I2C, UART
ADXL345	акселерометр	$\pm 2g$ – $\pm 16g$ / -	3.9 мг/LSB / -	40 мкА	I2C, SPI

2 РОЗРОБКА АПАРАТНОГО МОДУЛЯ

2.1 Структурна схема пристрою

Для реалізації проекту було запропоновано структурну схему, яка дає можливість визначати положення тіла у просторі, а також виводити на екран результати вимірювання. Запропонована схема наведена на рисунку 2.1.

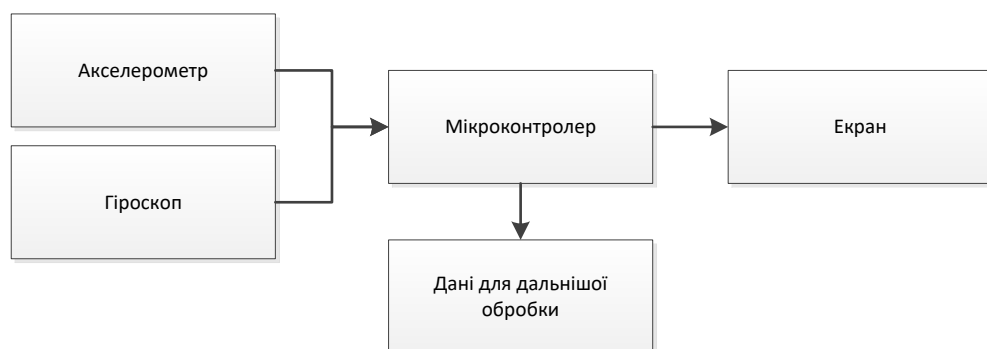


Рисунок 2.1 – Структурна схема проекту

У якості модуля керування використовується мікроконтролер, який зчитує інформацію з пристроїв визначення положення тіла, а саме гіроскопа та акселерометра, та виводить дані для подальшого оброблення, а також на екран.

2.2 Вибір мікроконтролера

Для керування запропонованим пристроєм було обрано плату Arduino UNO. Arduino Uno – це одна з найпопулярніших плат для розробки електронних проектів, яка використовує відкриту апаратну та програмну платформу. Arduino Uno має багатий функціонал і простий у використанні інтерфейс.

Плата базується на мікроконтролері ATmega328P та дозволяє підключати різні датчики, дисплеї, мотори, сенсори та інші компоненти для створення інтерактивних проектів. Для спілкування з зовні лата має 14 цифрових пінів, шість з них можна використовувати як ШІМ-виходи, також є в наявності шість аналогових пінів для підключення різноманітних датчиків. Arduino Uno може живитися як від USB, так і від зовнішнього джерела напруги, наприклад, від батареї або адаптера.

Програмування мікроконтролера можна виконувати через середовище Arduino IDE, яке доступне для Windows, macOS та Linux, мова програмування – це подібність до C/C++. Завдяки численним бібліотекам та спільноті розробників, використання Arduino Uno є дуже зручним для створення своїх проектів. Це дозволяє користувачам дуже швидко створювати свої проекти використовуючи готові скетчі, також модифікувати і вдосконалювати як апаратну частину, так і програмне забезпечення під свої потреби.

За допомогою спеціальних розширень, таких як додаткові плати поширення Shields можна легко підключити до Arduino багато пристроїв: від Wi-Fi модулів до моторів і сенсорів. Також Arduino дозволяє підключити різні модулі для обміну даними через інтернет, що відкриває можливості для створення "розумних" пристроїв та інтернету речей. Запропонований в роботі модуль можна буде використовувати у різноманітних таких проектах, де буде необхідним визначення положення тіла у просторі.

На платі Arduino Uno стоїть мікроконтролер ATmega328P, як зазначалось вище. Це популярний 8-розрядний чіп від компанії Atmel, на даний час це частина Microchip Technology. Цей мікроконтролер завоював популярність завдяки своїй простоті, доступності, хорошій продуктивності і широкому використанню в навчальних та аматорських проектах.

Основні характеристики ATmega328P – це 8-розрядний мікроконтролер, що працює на архітектурі AVR. Він має вбудовану програмовану пам'ять на основі Flash та EEPROM для зберігання даних і

програм. Обсяг Flash пам'яті складає 32 КБ для зберігання програм (з яких 0,5 КБ займає завантажувач), та SRAM (статичної оперативної пам'яті) – 2 КБ для тимчасового зберігання змінних даних. Також у складі мікроконтролера є EEPROM обсягом 1 КБ для зберігання даних, які не змінюються після відключення живлення (наприклад, налаштування або параметри). Тактування мікроконтролера здійснюється в стандартному режимі на частоті 16 МГц, але можна використовувати й інші частоти в залежності від зовнішніх генераторів чи осциляторів.

Для спілкування з зовнішніми пристроями (для підключення сенсорів, модулів, екранів та ін.) є в наявності 23 порти вводу/виводу, з них шість портів можуть працювати як ШІМ (широкоімпульсна модуляція), наприклад для регулювання швидкості мотора, яскравості LED тощо. Також мікроконтролера має шість аналогових входів для підключення аналогових сенсорів, наприклад, датчиків температури, освітленості тощо.

Також ATmega328P має наступні комунікаційні інтерфейси:

- USART (Universal Synchronous and Asynchronous Receiver-Transmitter) для серійного зв'язку через UART (з'єднання з комп'ютером, модулями Bluetooth, GPS тощо);

- SPI (Serial Peripheral Interface) для високошвидкісного обміну даними з периферійними пристроями, такими як карти пам'яті, датчики тощо;

- I2C (Inter-Integrated Circuit) для з'єднання з пристроями, які підтримують цей стандарт (датчики, дисплеї, RTC – годинники реального часу).

ATmega328P має у складі аналогово-цифровий перетворювач (ADC), а саме 10-розрядний АЦП з 6 аналоговими входами, що дозволяє конвертувати аналогові сигнали в цифрову форму для обробки. Також ATmega328P має кілька таймерів і лічильників, що дає можливість реалізувати точне управління часом, генерувати сигнали PWM або відліковувати час, або щось підраховувати. Вони використовуються, наприклад, для генерування звукових сигналів, управління моторами або вимірювання часу.

Мікроконтролер ATmega328P відомий низьким споживанням енергії, що робить його ідеальним для проектів, де важлива автономність, таких як роботи, датчики або портативні пристрої. Тобто ATmega328P – це універсальний і доступний мікроконтролер, який зручний для реалізації електронних проектів.

2.3 MPU-6050

На основі проведеного аналізу для визначення положення тіла був обраний датчик MPU-6050. MPU6050 – це потужний та універсальний сенсор руху, який поєднує в собі 3-осьовий гіроскоп та 3-осьовий акселерометр. Цей датчик широко використовується в проектах робототехніки, навігації та стабілізації.

MPU6050 – це інтегрована мікросхема, яка має наступні технічні характеристики:

- 3-осьовий MEMS гіроскоп;
- 3-осьовий MEMS акселерометр;
- цифровий процесор руху (DMP);
- 16-бітовий АЦП для кожного каналу;
- термодатчик;
- FIFO-буфер об'ємом 1024 байт;
- програмовані користувачем цифрові фільтри для гіроскопа, акселерометра та термодатчика;
- інтерфейс I2C для запису та читання регістрів пристрою, що працює на частоті до 400кГц.

Внутрішня блок-схема MPU-6050 наведена на рисунку 2.2.

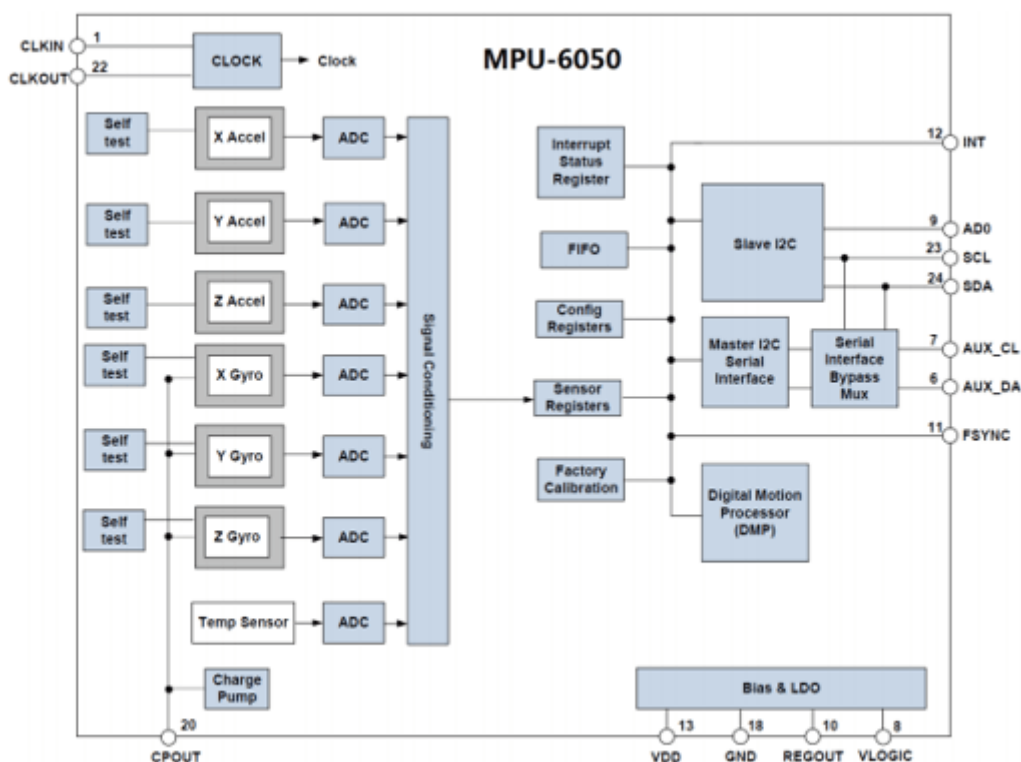


Рисунок 2.2 – Внутрішня блок-схема MPU-6050

Цей сенсор може вимірювати кутову швидкість та прискорення з високою точністю. Діапазони виміру можна налаштувати.

Гіроскоп вимірює кутову швидкість обертання навколо осі, яка виражається в радіанах на секунду (рад/с) або умовно в градусах/секунду ($^{\circ}/с$). Це важливо для відстеження обертання пристрою. Для знаходження поточного кута за швидкістю потрібно інтегрувати цю швидкість. Гіроскоп має 16-бітну роздільна здатність для кутової швидкості. Програмований діапазон вимірювань: ± 250 , ± 500 , ± 1000 , ± 2000 $^{\circ}/сек$, діапазон обирають залежно від точності та діапазону кутових швидкостей.

Акселерометр вимірює лінійне прискорення вздовж осі, зазвичай у метрах на секунду у квадраті ($м/с^2$) або g (де $1g = 9.81$ $м/с^2$). Він використовується для визначення кутів нахилу, орієнтації та прискорень. Якщо датчик лежить на столі або рухається з постійною швидкістю, то на осі буде спроектовано вектор сили тяжіння. Якщо датчик рухається з

прискоренням – до прискорення вільного падіння отримаємо складові вектора прискорення. Якщо датчик знаходиться у вільному падінні (у тому числі на орбіті планети) – величини прискорень по всіх осях дорівнюватимуть 0. Знаючи проекції вектора сили тяжіння можна з високою точністю визначити кут нахилу датчика щодо нього. Програмований діапазон вимірювань: $\pm 2g$, $\pm 4g$, $\pm 8g$, $\pm 16g$, його можна вибрати залежно від потрібної чутливості. Роздільна здатність акселерометра 16 біт, що дозволяє точно вимірювати прискорення.

MPU6050 має розширені можливості, а саме багато додаткових функцій:

1) DMP (Digital Motion Processor), що дозволяє виконувати складні обчислення безпосередньо на чипі.

2) температурний сенсор, за допомогою якого можна отримати дані про температуру;

3) низькоспоживаючі режими для економії енергії у мобільних проектах.

4) можна налаштувати сенсор для генерації переривань за певних подій руху.

Датчик MPU-6050 можна підключати за допомогою наступних інтерфейсів:

1) I2C інтерфейс (Inter-Integrated Circuit) йде за замовчуванням. MPU6050 використовує інтерфейс I2C для зв'язку з мікроконтролером. Це стандартний інтерфейс зв'язку, який дозволяє підключати декілька пристроїв із мінімальною кількістю проводів. Він використовує дві лінії: SDA (Data) передачі даних і SCL (Clock) для синхронізації. Це найбільш поширений інтерфейс для роботи з MPU-6050, так як він дозволяє легко підключити датчик до мікроконтролера, наприклад Arduino або Raspberry Pi. Швидкість передачі даних через I2C до 400 кбіт/с (для режиму Fast Mode).

2) SPI (Serial Peripheral Interface) хоча не є стандартним інтерфейсом для MPU-6050, цей датчик також підтримує його. SPI вимагає більше

проводів (зазвичай 4), але може запропонувати більш високу швидкість передачі в порівнянні з I2C.

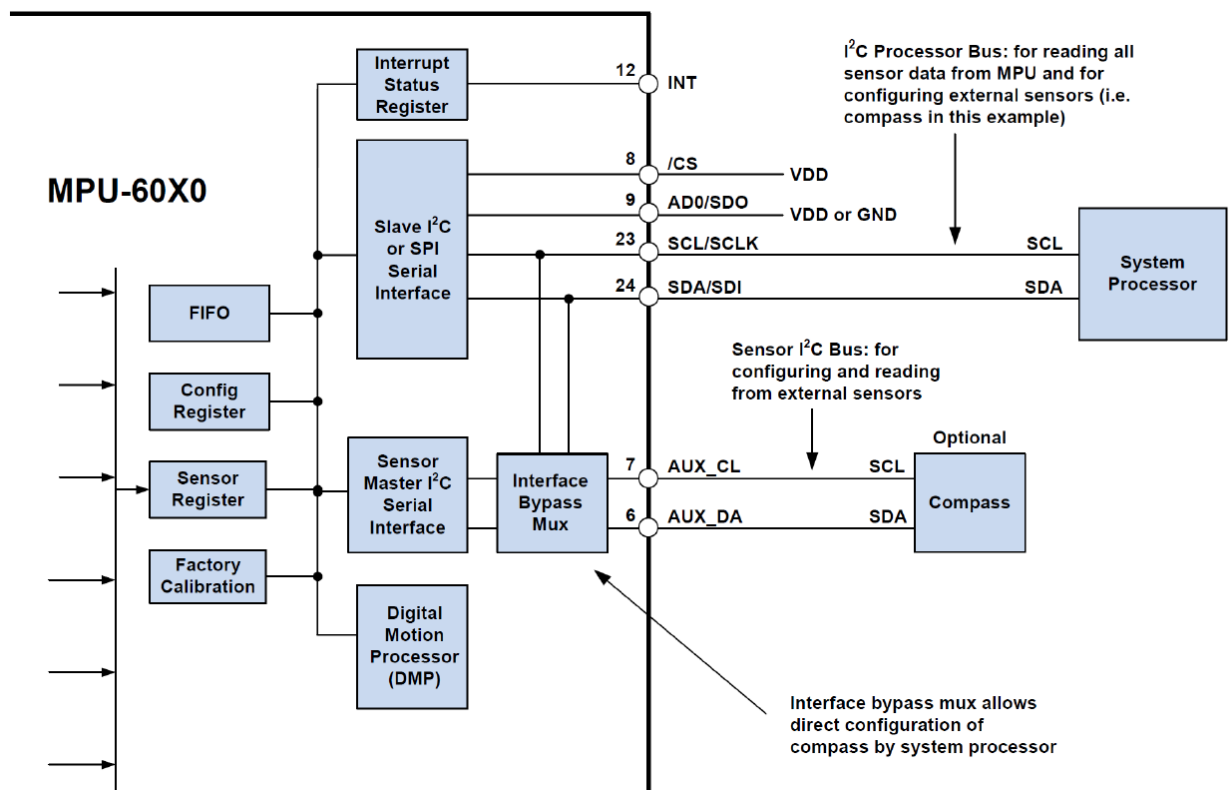


Рисунок 2.3 – Підключення MPU-6050

Також даний пристрій має вбудований температурний датчик, який може бути корисним для корекції даних акселерометра і гіроскопа, а також для моніторингу температурних змін.

Вбудований цифровий фільтр низьких частот (DLPF) використовується для згладжування вихідних даних акселерометра та гіроскопа. Це допомагає зменшити шум та коливання, покращуючи точність вимірювань.

2.4 OLED-дисплей

Для виведення інформації в роботі було використано дисплей Monochrome 128x64 with I2C interface. Монохромний OLED-дисплей із

роздільною здатністю 128x64 пікселів. Він є популярним графічним дисплеєм, який використовується в різноманітних електронних проєктах і пристроях із обмеженим розміром.

OLED-дисплей складається з 128 стовпців і 64 рядків пікселів, де кожен піксель може бути або увімкнений (світиться), або вимкнений (не світиться). Завдяки органічним світлодіодам (OLED – Organic Light Emitting Diode), дисплей не потребує підсвічування, оскільки кожен піксель світиться самостійно.

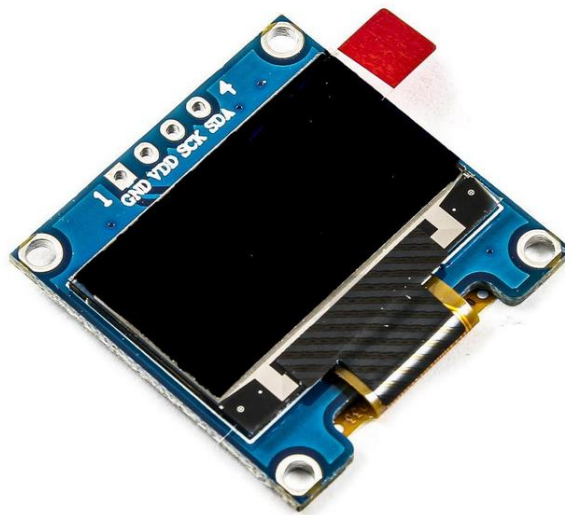


Рисунок 2.4 – OLED-дисплей

Дисплей має високу контрастність, широкий кут огляду та швидкий час відгуку. Завдяки цим характеристикам даний дисплей є зручним для відображення як текстової, так і графічної інформації. Він зручний навіть у складних умовах освітлення. Енергоспоживання такого дисплея дуже низьке, особливо якщо виводиться переважно чорний фон з мінімальною кількістю активних пікселів, що робить його придатним для пристроїв на батарейному живленні.

Керування дисплеєм зазвичай здійснюється через інтерфейси I²C або SPI, залежно від конкретної моделі. Найпоширеніший контролер – SSD1306, який підтримується багатьма бібліотеками для мікроконтролерів (наприклад,

Arduino, STM32, ESP). Завдяки цьому дисплей легко інтегрується у вбудовані системи для відображення меню, даних з датчиків, графіків, значків тощо.

Для інтеграції дисплея в мікроконтролерну систему зазвичай використовують інтерфейс I²C, який потребує всього двох сигнальних ліній – SDA (дані) та SCL (тактова частота). Це дозволяє значно зменшити кількість потрібних виводів і спростити підключення. Для роботи дисплею необхідно підключити живлення або 3.3В або 5В. Більшість дисплеїв SSD1306 сумісні як з 3.3В, так і з 5В логікою.

3 РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ

Для визначення положення тіла у просторі було розроблено алгоритм роботи, який наведено на рисунку 3.1.

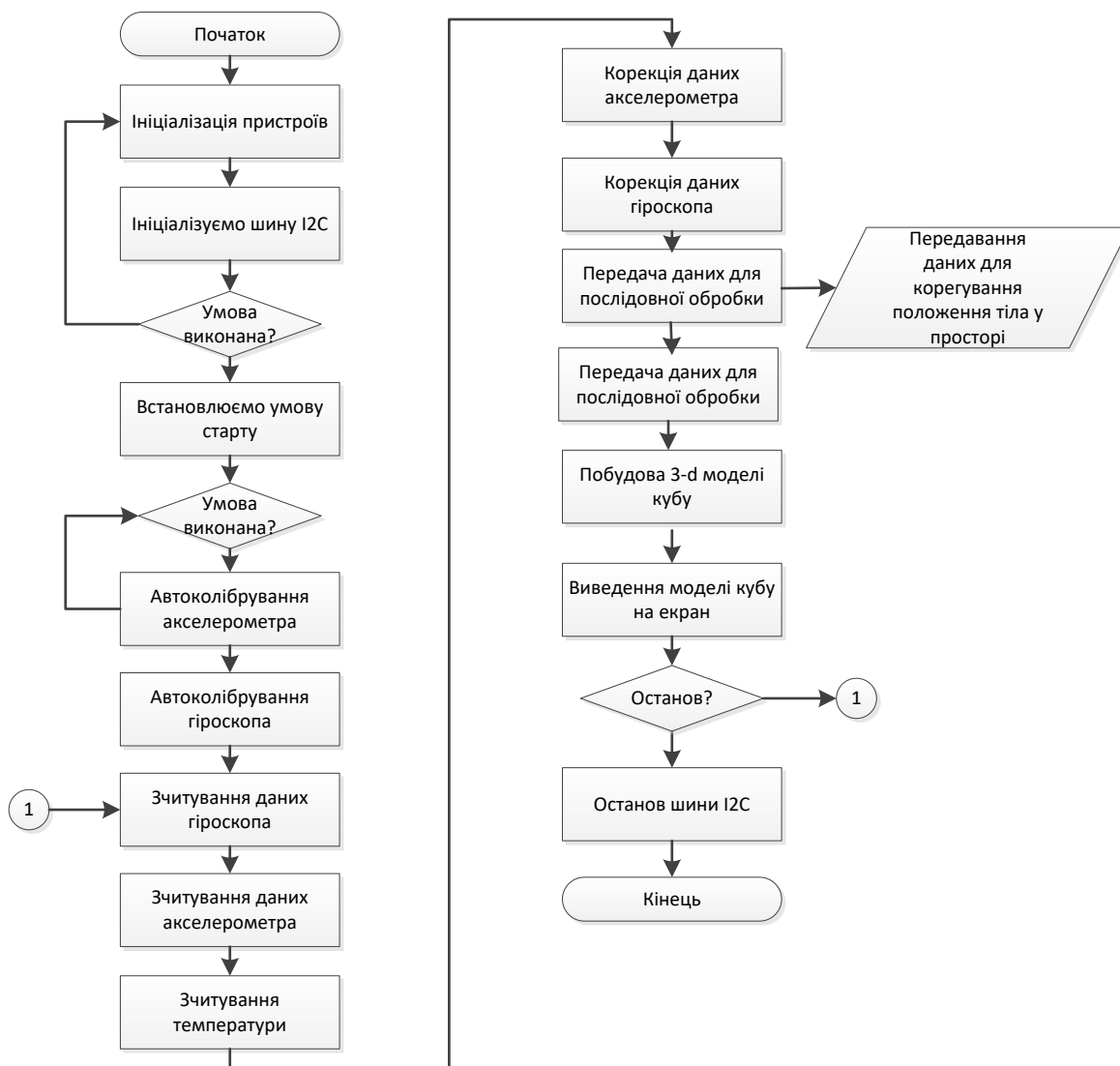


Рисунок 3.1 – Алгоритм роботи пристрою

Алгоритм включає в себе наступну послідовність. Спочатку виконується ініціалізація необхідних пристроїв. Після ініціалізації стартує шина I2C та зчитується готовність пристрою до роботи. Після відбувається

актоколібрування спочатку гіроскопу, потім акселерометру. Після зчитуються дані, корегуються відповідно погрішностей та виводяться для подальшого використання. Для того, щоб побачити начно, як відбувається процес читання, було в розробці було використано виведення даних на екран для управління 3-d моделлю куба. Для відтворення моделі куба було використано проект [14]. Відповідно до запропонованого алгоритму було реалізовано програму керування.

В проєкті були використані наступні бібліотеки:

Лістинг 3.1

```
#include <Wire.h>
#include <Adafruit_SSD1306.h>
#include <Adafruit_GFX.h>
```

<Wire.h> бібліотека дає можливість працювати з інтерфейсом I2C, який необхідний для зв'язку з датчиками та екранами, такими як MPU-6050 та OLED дисплеї. Вона дозволяє обмінюватися даними між мікроконтролером та пристроями через дві лінії (SDA та SCL).

<Adafruit_SSD1306.h> бібліотека використовується для роботи з OLED-дисплеями, які побудовані на контролері SSD1306. Ці дисплеї часто використовуються для відображення текстової та графічної інформації та підключаються через інтерфейс I2C або SPI.

<Adafruit_GFX.h> бібліотека необхідна для роботи з графічними дисплеями, а саме SSD1306. Вона надає функції відрисовки ліній, тексту, прямокутників та інших графічних елементів на екрані.

Були створені відповідні змінні для читання даних з акселерометра, темподатчика та гіроскопа, та змінні для збереження даних автоколібрування. Ці змінні наведені у лістингу 3.2.

Лістинг 3.2

```
int16_t ax, ay, az;
int16_t temp;
```

```
int16_t gx, gy, gz;

float ax_offset = 0, ay_offset = 0, az_offset = 0;
float gx_offset = 0, gy_offset = 0, gz_offset = 0;

float xByGyro, yByGyro, zByGyro;
```

Також був оголошений масив для зберігання координат 3d-куба, який наведено у Лістингу 3.3 [14].

Лістинг 3.3

```
const int cube1 [3][3] =
{
  { -20, -20, 20 },      // x, y, z
  { 20, -20, 20 },
  { 20, 20, 20 },
  { -20, 20, 20 },
  { -20, -20, -20 },
  { 20, -20, -20 },
  { 20, 20, -20 },
  { -20, 20, -20 }
};
```

Для того, щоб система почала працювати, необхідно виконати ініціалізацію всіх пристроїв. У лістингу наведено ініціалізацію дисплею, MPU-6050, а також UART. Інтерфейс UART потрібен для того, щоб виводити данні, отримані від гіроскопу та акселерометру в монітор порту на екран або комп'ютера, або в поле монітору порта стимулятора. Ініціалізація інтерфейсу I2C відбувається для висилання даних на екран OLED дисплей. При ініціалізації задаються такі параметри як адресу пристрою, розмір екрану, розмір шрифту та його колір та інше.

Також на цьому етапі відбувається виклик функції калібрування акселерометра та гіроскопа для виконання подальшої корекції даних.

Лістинг 3.4

```
//Ініціалізація простоїв

void setup() {
  Wire.begin(); // Ініціалізуємо I2C
```

```

Serial.begin(9600); // Ініціалізуємо серіальну комунікацію

display.begin(SSD1306_SWITCHCAPVCC, 0x3C); // Вказуємо
адресу пристрою на шині
display.clearDisplay();
display.setTextSize(2); // Вказуємо розмір шрифту дисплею
display.setTextColor(SSD1306_WHITE); // вказуємо колір
напису

display.setCursor(30, 10);
display.println("Kharkiv");
display.display();
delay(3000);
// display.clearDisplay(); // Очищаємо екран
delay (100); // Даємо трохи часу пристроям на синхронізацію

// Ініціалізуємо MPU-6050
Wire.beginTransmission(MPU_addr); // Починаємо передачу до
MPU-6050
byte testConnection = Wire.endTransmission(); // Завершуємо
та перевіряємо відповідь

// Перевіряємо підключення до MPU-6050
Serial.println(testConnection ? "Помилка підключення до
MPU-6050" : "MPU-6050 connected!"); // Перевірка з'єднання
// Виклик функції автокалібрування для акселерометра
calibrateAcceleration();
// виклик функції автокалібрування для гіроскопа
calibrateGyroscope();

}

```

Функція автокалібрування акселерометра та гіроскопа наведені у Лістингу 3.5. Автокалібрування – це процес усунення апаратних похибок датчика, таких як зміщення нуля (bias). Вона виконується програмно шляхом усереднення безлічі вимірювань при нерухомому датчику і подальшого віднімання цих значень всіх подальших показань. Для цього спочатку виконують ініціалізацію датчиків та перевіряють його доступність, а потім збирають кілька сотень вимірювань, коли MPU-6050 лежить нерухомо. Для цього витримується невелика затримка, та датчик MPU-6050 повинен знаходитися абсолютно нерухомо на рівній та стабільній поверхні. У процесі автокалібрування обчислюють середнє значення для кожного з параметрів (X, Y, Z для акселерометра та гіроскопа) та використовують ці значення як

поправки (offset'и). Після цього кожен новий вимір коригується шляхом віднімання розрахованих зсувів, що значно підвищує точність. Щоб не виконувати калібрування під час кожного запуску, отримані дані можна зберегти в EEPROM. Такий метод дозволяє мінімізувати помилки вимірювань та покращити роботу системи стабілізації чи навігації.

Лістинг 3.5

```
// виклик функції автоколібрування для гіроскопа
void calibrateGyroscope() {
    int samples = 200;
    long gx_sum = 0, gy_sum = 0, gz_sum = 0;

    for (int i = 0; i < samples; i++) {
        Wire.beginTransmission(MPU_addr);
        Wire.write(0x43); // Починаємо з регістра GYRO_XOUT_H
        Wire.endTransmission(false);
        Wire.requestFrom(MPU_addr, 6, true);

        gx = Wire.read() << 8 | Wire.read();
        gy = Wire.read() << 8 | Wire.read();
        gz = Wire.read() << 8 | Wire.read();

        gx_sum += gx;
        gy_sum += gy;
        gz_sum += gz;
        delay(10);
    }

    gx_offset = gx_sum/samples;
    gy_offset = gy_sum/samples;
    gz_offset = gz_sum/samples;

    Serial.println("Calibration complete");
    Serial.print("gx_offset:"); Serial.println(gx_offset);
    Serial.print("gy_offset:"); Serial.println(gy_offset);
    Serial.print("gz_offset:"); Serial.println(gz_offset);
}

// Автоколібрування акселерометра
void calibrateAcceleration() {
    int samples = 200;
    long ax_sum = 0, ay_sum = 0, az_sum = 0;

    for (int i = 0; i < samples; i++) {
        Wire.beginTransmission(MPU_addr);
        Wire.write(0x3B); // Починаємо з регістру ACCEL_XOUT_H
        Wire.endTransmission(false);
```

```

Wire.requestFrom(MPU_addr, 6, true);

ax = Wire.read() << 8 | Wire.read();
ay = Wire.read() << 8 | Wire.read();
az = Wire.read() << 8 | Wire.read();

ax_sum += ax;
ay_sum += ay;
az_sum += az;
delay(10);
}

ax_offset = ax_sum/samples;
ay_offset = ay_sum/samples;
az_offset = az_sum/samples;

Serial.println("Calibration complete");
Serial.print("ax_offset:"); Serial.println(gx_offset);
Serial.print("ay_offset:"); Serial.println(gy_offset);
Serial.print("az_offset:"); Serial.println(gz_offset);
}

```

Функція читання «сирих» даних наведена у лістингу 3.6. Ця функція здійснює читання даних з MPU-6050 через інтерфейс I2C, починаючи з регістра `0x3B` (відповідно до документації на пристрій), який містить старший байт акселерометра по осі X. Спочатку вона встановлює з'єднання з датчиком, вказуючи його адресу, а потім відправляє команду на вибір потрібного регістра. Після цього передача завершується без звільнення шини, щоб одразу запросити 14 байтів даних. MPU-6050 послідовно відправляє значення акселерометра осях X, Y і Z, потім показання вбудованого температурного датчика, а після цього дані гіроскопа. Кожне значення передається у двох байтах (старший та молодший), які об'єднуються у 16-бітові змінні. Далі ці дані передаються в головну програму для подубови 3d-моделі кубу та для виведення у монітор послідовного порта. Також вони можуть поступати для подальшого коректування положення тіла у просторі та його стабілізації.

Формат передачі даних акселерометра наступний: дані акселерометра для осі X, осі Y, осі Z займають по два байти для кожної осі. Ці два байти

передаються у форматі big-endian (старший байт спочатку), тобто старший байт йде першим, потім молодший байт.

Структура даних акселерометра, яка передається від MPU6050 наведена у таблиці 3.1.

Таблиця 3.1 – Структура даних акселерометра

Регістри	Розмір	Дані	Опис
0x3B - 0x40	6 байт	X-axis (2 байти)	Дані акселерометра по осі X
0x41 - 0x44	6 байт	Y-axis (2 байти)	Дані акселерометра по осі Y
0x45 - 0x48	6 байт	Z-axis (2 байти)	Дані акселерометра по осі Z

Після того, як ці байти отримані, їх можна об'єднати в 16-бітові значення для кожної осі. Наприклад, для осі X:

```
ax = Wire.read() << 8 | Wire.read();
```

Також необхідно врахувати, що дані являють собою 16-бітові значення, які можуть бути як позитивними, так і негативними. Якщо акселерометр працює в діапазоні ****±2g****, дані варіюватимуться в межах від -32768 до 32767 у 16-бітному форматі.

Діапазони для акселерометра можуть бути налаштовані (±2g, ±4g, ±8g, ±16g), і значення даних масштабуватимуться залежно від вибраного діапазону. Для правильної інтерпретації отриманих значень потрібно враховувати коефіцієнти масштабування кожного діапазону акселерометра.

Лістинг 3.6

```
void readData() {
```

```

Wire.beginTransaction(MPU_addr);
Wire.write(0x3B); // Начинаяем с регистра ACCEL_XOUT_H
Wire.endTransmission(false);
Wire.requestFrom(MPU_addr, 14, true); // Читаем 14 байтов

ax = Wire.read() << 8 | Wire.read();
ay = Wire.read() << 8 | Wire.read();
az = Wire.read() << 8 | Wire.read();
temp = Wire.read() << 8 | Wire.read();
gx = Wire.read() << 8 | Wire.read();
gy = Wire.read() << 8 | Wire.read();
gz = Wire.read() << 8 | Wire.read();

}

```

В головній функції `loop()` (лістинг 3.7) виконується кілька ключових завдань, пов'язаних з обробкою даних MPU-6050 та їх візуалізацією.

Він спочатку викликає функцію `readData()`, яка зчитує дані про прискорення, кутові швидкості і температуру, заповнюючи глобальні змінні `ax, ay, az, gx, gy, gz, temp`. Для цього мікроконтролер посилає команду для читання даних із пристрою у відповідь MPU6050 відправляє 6 байт даних, які включають:

- 2 байти для осі X акселерометра,
- 2 байти для осі Y акселерометра,
- 2 байти для осі Z акселерометра,
- 2 байти для осі X гіроскопа,
- 2 байти для осі Y гіроскопа,
- 2 байти для осі Z гіроскопа.

Після процесу зчитування даних, вони перетворюються на реальні фізичні величини (гравітаційні одиниці для акселерометра та градуси за секунду для гіроскопа). Дані від акселерометра в MPU6050 передаються по інтерфейсу I2C у вигляді послідовності байтів. Зокрема, для отримання даних від акселерометра (по осях X, Y і Z), потрібно вважати 6 байт інформації, які будуть містити 16-бітні дані для кожної з осей. Кожен із цих байтів буде

передаватися в тому порядку, в якому вони зберігаються в пам'яті мікросхеми.

Далі дані надходять в обробку, де прискорення переводиться із сирих значень у метри на секунду у квадраті, а температура – у градуси Цельсія. Акселерометричні значення осей X, Y і Z перетворюються на прискорення m/s^2 , помножуючи їх на 9.81 після масштабування. Далі розраховується температура у градусах Цельсія, використовуючи формулу $(temp/340.0) + 36.53$.

Після цього дані гіроскопа наводяться до кутових швидкостей градусах в секунду, використовуючи коефіцієнт масштабування 131. Потім відбувається інтегрування значень гіроскопа для визначення кутів повороту пристрою, і вони комбінуються з кутами, обчисленими на основі даних акселерометра.

Щоб кути залишалися у допустимому діапазоні, вони коригуються так, щоб перебувати в межах від 0 до 2π . Потім ці кути використовуються для обертання віртуального 3D-об'єкта (а саме, куба), де поворот відбувається послідовно по осях X, Y і Z. Після цього координати вершин коригуються з урахуванням перспективи, центруються на екрані, і оновлені значення передаються масив для відтворення.

У фінальній частині викликається `draw_wireframe()`, що відповідає за візуалізацію об'єкта на екрані. Цей код фактично реалізує простий тривимірний трекер руху, що візуалізує орієнтацію пристрою у просторі.

Лістинг 3.7

```
void loop() {
    readData(); // Отримання даних з сенсора

    float acceleration_x = (ax / 163840.0);
    float acceleration_x_ms2 = acceleration_x * 9.81;
    float acceleration_y = (ay / 163840.0);
    float acceleration_y_ms2 = acceleration_y * 9.81;
```

```

float acceleration_z = (az / 163840.0);
float acceleration_z_ms2 = acceleration_z * 9.81;

Serial.print("a/g:\t");
Serial.print(acceleration_x_ms2); Serial.print("\t");
Serial.print(acceleration_y_ms2); Serial.print("\t");
Serial.print(acceleration_z_ms2);
Serial.print("\t");Serial.print("\t");

//Температура (°C)
float temp_f = (temp / 340.0) + 36.53;

Serial.print(temp_f);
Serial.print("\t");Serial.print("\t");

float scaleFactorGyroscope = 131;
float gyroscope_x = (gx - gx_offset ) /
scaleFactorGyroscope;
float gyroscope_y = (gy - gy_offset) /
scaleFactorGyroscope;
float gyroscope_z = (gz - gz_offset) /
scaleFactorGyroscope;

Serial.print(gyroscope_x); Serial.print("\t");
Serial.print(gyroscope_y); Serial.print("\t");
Serial.println(gyroscope_z);

delay(500); // затримка

float xByAccel = (float) ax * 0.0001;
float yByAccel = (float) ay * 0.0001;
float zByAccel = (float) az * 0.0001;

xByGyro += (float) gx * 0.00001;
yByGyro += (float) gy * 0.00001;
zByGyro += (float) gz * 0.00001;

float x = xByAccel + xByGyro;
float y = yByAccel + yByGyro;
float z = zByAccel + zByGyro;

```

Далі використовуючі приклади [14] відбувається відрисовування 3-d моделі кубу. Цей код виконує 3-d перетворення вершин куба. Спочатку він обертає вершини навколо осей X, Y та Z, використовуючи тригонометричні формули. Потім додає ефект перспективи, зменшуючи дальні точки, та зміщує об'єкт до центру екрана. Наприкінці оновлені координати

записуються до масиву `wireframe`, після чого викликається функція `draw_wireframe()`, яка виводить на екран оновлене зображення.

4 ТЕСТУВАННЯ ПРИСТРОЮ

Тестування пристрою проводилось в декілька етапів. Спочатку було проведено тестування на симуляторі Wokwi. В ході цього експерименту було перевірено процес калібрування пристрою. На рисунку 4.1 показаний проект на базі Arduino Uno, в якому використовується модуль MPU6050 (акселерометр та гіроскоп) та OLED-дисплей. Результати калібрування виводяться в консоль стимулятора. У верхній частині проекту можна побачити повзунки, які імітують реальні фізичні сили (зазвичай – гравітацію). Ці дані використовуються симулятором як реальні та відповідають положенню пристрою в просторі.

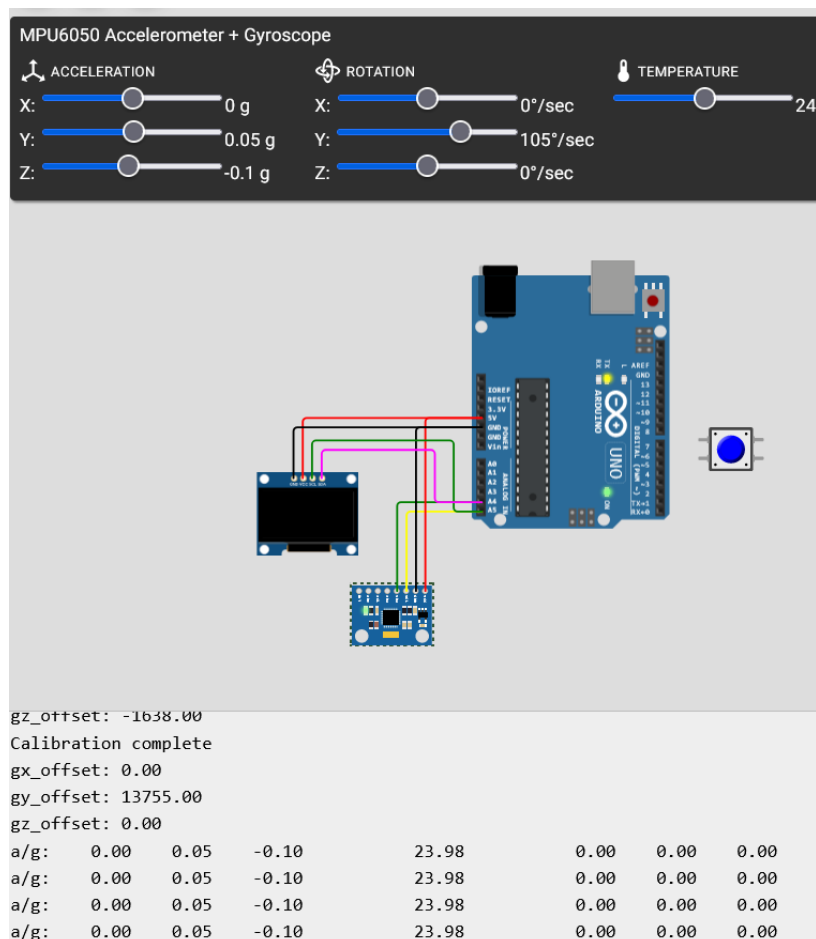


Рисунок 4.1 – Процес актоколібування

Процес авто калібрування відбувається наступним чином. Під час запуску виконується калібрування гіроскопа, а саме протягом кількох секунд модуль збирає дані при нерухомому положенні. За допомогою повзунків встановлюється початковий нахил тіла. Обчислюється середнє зміщення (offset) за кожним із трьох осей (X, Y, Z). Ці усунення зберігаються і потім віднімаються з усіх майбутніх вимірів.

На зображенні видно таке:

Calibration complete

gx_offset: 0.00

gy_offset: 13755.00

gz_offset: 0.00

А саме «gy_offset: 13755.00» означає, що по осі Y гіроскоп давав значне зміщення, яке тепер компенсуватиметься. Потім ця помилка вноситься для корегування зчитуємих даних. Це можна побачити на рисунку 4.2.

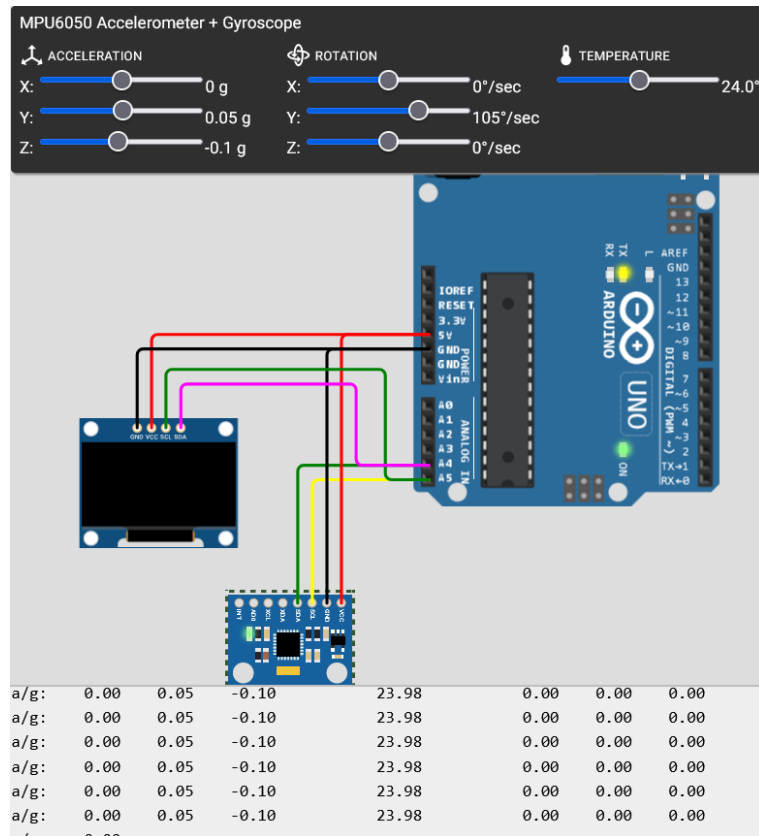


Рисунок 4.2 – Компенсація помилки після автокалібрування

На другому етапі було створено макет пристрою. Було проведено тестування макету для оцінки місцезнаходження тіла в просторі за допомогою інерційного датчика MPU 6050. Результати роботи можна побачити на наступних рисунках.

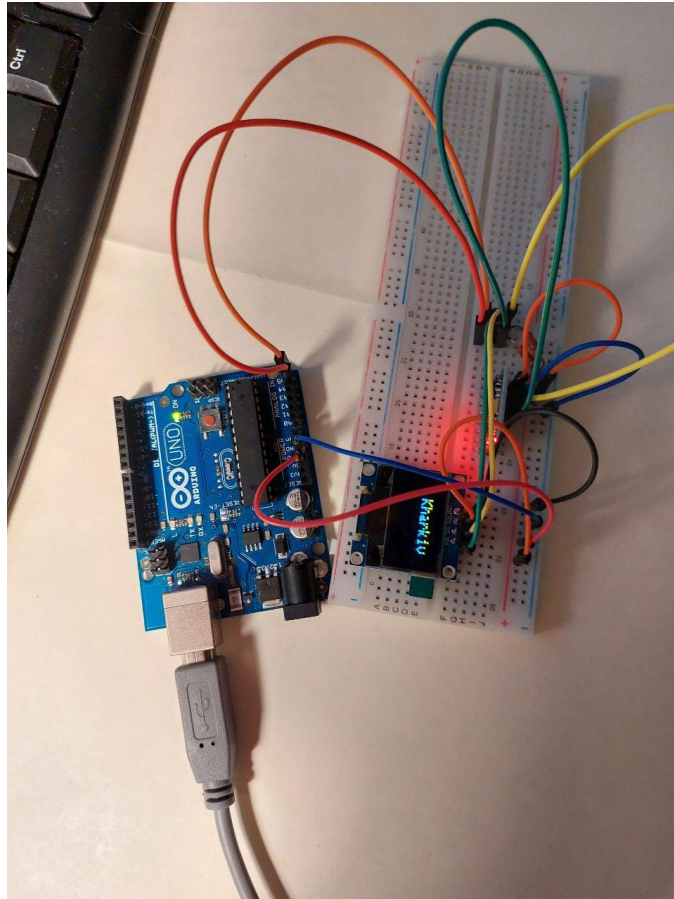


Рисунок 4.3 – Ініціалізація дисплея

На рисунку 4.3 наведено ініціалізацію дисплея. Спочатку на дисплей виводиться надпис «Kharkiv», після цього відбувається ініціалізація MPU 6050, а також проходить процес авто калібрування пристрою. Після цього на екрані з'являється зображення кубу, який через невелику затримку починає рух відповідно до положення тіла, на якому закріплений датчик.

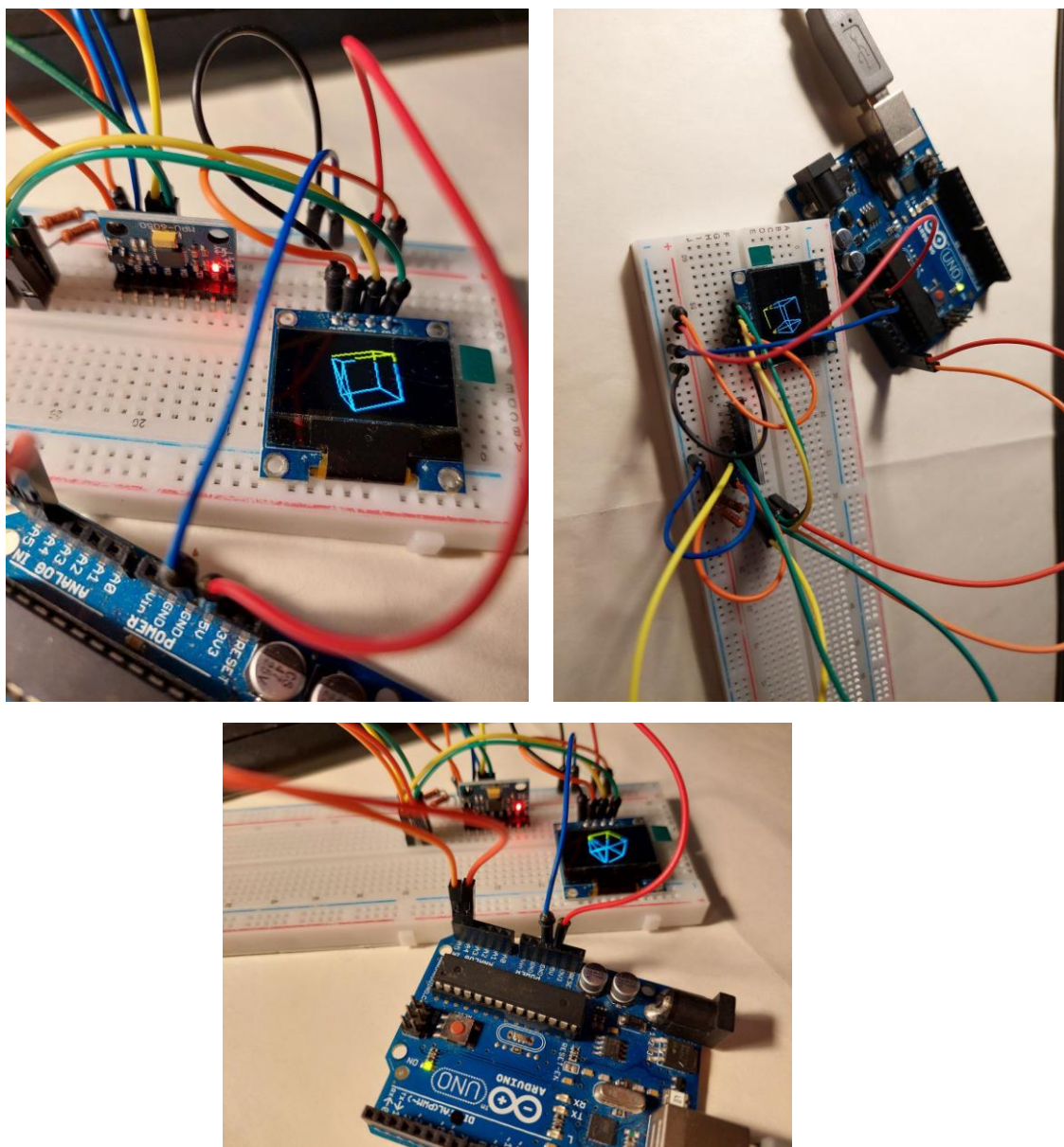


Рисунок 4.4 – Робота пристрою

На рисунку 4.4 можна побачити наступне. В ході експерименту фізичний модуль MPU-6050 був підключений Arduino Uno через COM-порт до комп'ютера. Після ініціалізації було запущено автоматичне калібрування, при якому визначилися усунення помилки гіроскопа. Акселерометр відкалібрувався з нульовими зсувами. Далі в реальному часі відбувалася передача та відображення даних акселерометра та гіроскопа Serial Monitor з частотою, що визначається в скетчі. Це дозволяє візуально контролювати рухи та повороти пристрою у просторі.

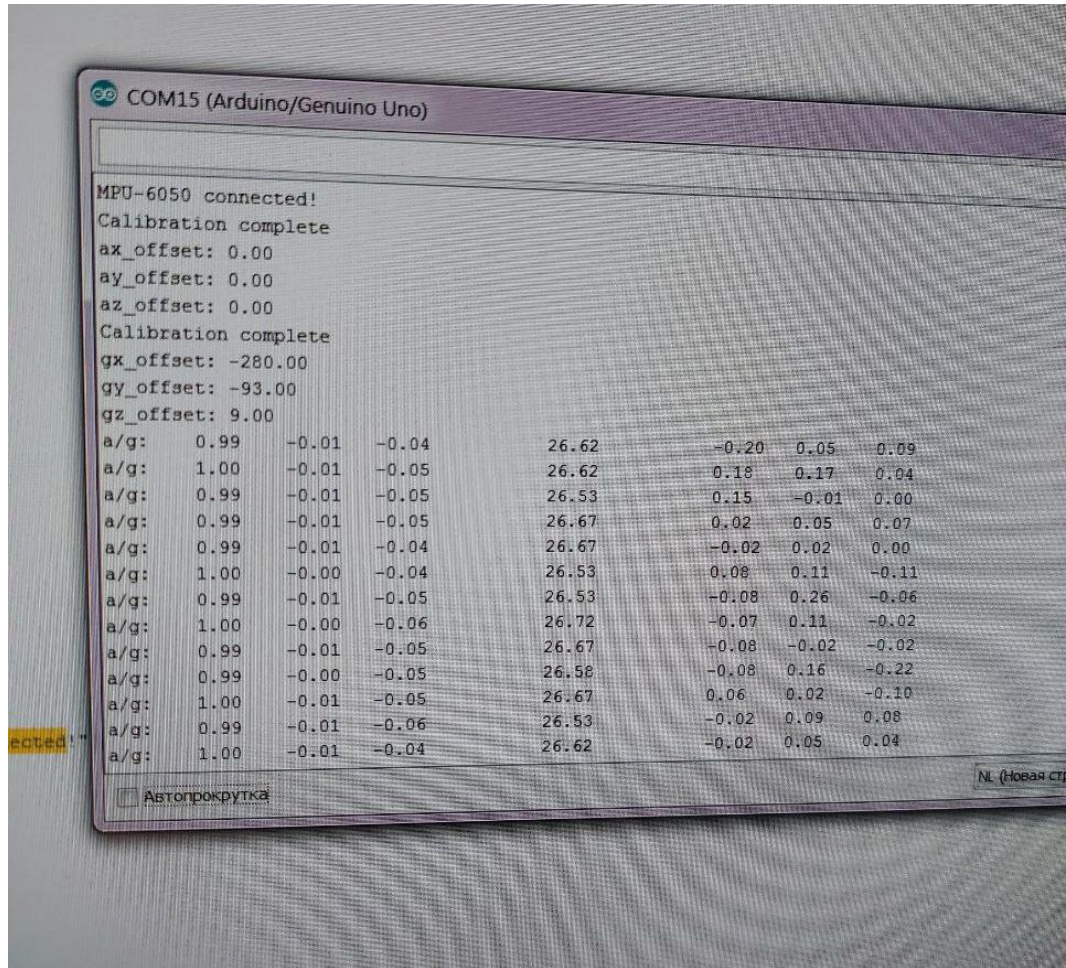


Рисунок 4.5 – Работа прибора

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було розроблено програмно-апаратний модуль для оцінки місцезнаходження тіла в просторі за допомогою інерційного датчика MPU 6050

Для цього було проведено аналіз існуючих модулів, розглянуто принцип їх роботи. Також проведено порівняльну характеристику і обрано необхідний пристрій для реалізації проекту.

Було розроблено структурну схему пристрою. Обрано відповідний мікроконтролер для реалізації програми управління.

Розроблено програмний модуль проекту, а саме алгоритм роботи пристрою та окремі програмні модулі. Проведено тестування пристрою.

Результати тестування показали, що модуль працює відповідно до технічного завдання, процес автокалібрування проходить успішно, помилки враховуються при визначенні положення тіла у просторі.

Даний модуль може використовуватися як окремий, так і для подальшого застосування у різноманітних розробках, де необхідно знати координати тіла у просторі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Матвієнко М. Комп'ютерна схемотехніка. Навчальний посібник / Матвієнко М., Розен В. – К. : Ліра-К, 2014. – 192 с.
2. Савченко О. М. Arduino: Створюємо свої проекти. Київ: Видавництво "Національний університет оборони України імені Івана Черняхівського", 2020. 208 с.
3. Pong P. Chu. FPGA Prototyping by Verilog Examples: Xilinx Spartan-3 Version, 1st edition / Pong P. Chu. – Wiley: Wiley-Interscience, 2008. – 528 с.
4. Sarah Harris. Digital Design and Computer Architecture: ARM Edition, 1st edition / S. Harris, D. Harris. – Morgan Kaufmann, 2015. – 584 с.
5. Ghlukhov, O. V., Kravchuk, O. O. & Levchenko, Je. V. (2020). Creation of a laboratory workshop based on the Arduino platform and its role in teaching students of technical universities of all forms of education in the specialty "Electronics". In Proceedings of the eleventh international scientific-practical conference «Prospects for the development of modern science and education», Lviv, 15-16 June, 2020: 13, Lviv
6. Кривуля Г. Ф. Схемотехніка: Навчальний посібник / Кривуля Г. Ф., Рябенський В. М., Буряк В. С. – Харків : ТОВ "Компанія СМІТ", 2007. – 250 с.
7. Тонкошкур О. С. Мікроконтролерні пристрої: навч. посіб. для студ. спец. «Мікро- та наноелектроніка», [Текст] / О. С. Тонкошкур, І. В. Гомілко, О. В. Коваленко. Дніпропетровський нац. ун-т ім. О. Гончара. – Д.: Вид-во ДНУ, 2011. – 264 с.
8. Stephen M. Trimberger. Field-Programmable Gate Array Technology / Stephen M. T. – Springer, 1994. – 273 с.

9. Харпер Р. Всередині розумного будинку: ідеї, можливості та методи. Harper, R. Inside the smart home: Ideas, possibilities and methods. in Richard Harper Inside the smart home. / Р. Харпер // Нью Йорк. – 2003, С.1-14
10. Wayne Wolf. FPGA-Based System Design, Prentice Hall Modern Semiconductor Design Series / Wolf W. – Prentice Hall, 2004. – 530 с.
11. ДСТУ ГОСТ 7.1:2006. Система стандартів з інформації, бібліотечної та видавничої справи. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання / Нац. стандарт України. – Вид. офіц. – [Чинний від 2007-07-01]. – К.: Держспоживстандарт України, 2007. – 47 с.
12. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання / Нац. стандарт України. – Вид. офіц. – [Уведено вперше ; чинний від 2016-07-01]. – К.: ДП «УкрНДНЦ», 2016. – 17 с.
13. Analysis of the implementation efficiency of digital signal processing systems on the technological platform SoC Zynq 7000 / Olexander Shkil, Oleh Filippenko, Dariia Rakhlis, Inna Filippenko, Valentyn Kornienko // Radioelectronic and Computer Systems. – 2024. – No. 4 (112). – P. 168-177.
14. <http://colinord.blogspot.com/2015/01/arduino-oled-module-with-3d-demo.html/> 11.05.2025 – Загол. з екрану
15. Evaluation of the performance of a computing cluster based on single board raspberry pi 3b+ computer / O. Barkovska, V Korniienko, I Filippenko et al. // 4th KhPI Week on Advanced Technology (KhPIWeek), Kharkiv, Ukraine, 02-06 October, 2023: proceedings. – P. 1–6.