

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
«__» _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА
**АПАРАТНО-ПРОГРАМНИЙ КОМПЛЕКС ДЛЯ REAL-
TIME СИМУЛЯЦІЇ РУХУ ОБ'ЄКТУ НА БАЗІ
ГІРОСКОПІЧНОГО ДАТЧИКА MPU-6050**

Спеціальність 123 Комп'ютерна інженерія
123 – КМР.1 – 605М.21717118

Студент

_____ Р. А. Чернявський
підпис
«__» _____ 2023р.

Керівник канд. фіз.–мат. наук, доцент

_____ С. В. Пузирьов
підпис
«__» _____ 2023р.

Миколаїв – 2023

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	4
ВСТУП	5
1 огляд ІСНУЮЧИХ РІШЕНЬ СИСТЕМИ, ЩО РОЗРОБЛЯЄТЬСЯ. ФОРМУВАННЯ ВИМОГ ДО АПАРАТНО-ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	7
1.1 Методи орієнтації у просторі	7
1.2 Вимоги до системи	19
1.3 Висновки до розділу	19
2. АПАРАТНО-ПРОГРАМНИХ КОМПОНЕНТІВ	20
2.1 Огляд компонентів апаратно-програмного комплексу	20
2.2 Опис програмного середовища Fritzing	33
Висновки до розділу 2	36
3 РОЗРОБКА АПАРАТНО-ПРОГРАМНОГО КОМПЛЕКСУ	37
3.1 Алгоритм роботи системи real time симуляція руху об'єкту	37
3.2 Створення блок-схеми алгоритму визначення переміщення	37
3.3 Проектування системи real time симуляції руху об'єкту на базі гіроскопічного датчика MPU-6050	38
3.4 Проектування концептуальної схеми апаратно-програмного комплексу	39
3.5 Розробка структурної діаграми схеми апаратно-програмного комплексу	42

Кафедра комп'ютерної інженерії		
Апаратно-програмний комплекс для real-time симуляції руху об'єкту на базі гіроскопічного датчика MPU-6050		
3.6	Розробка діаграми послідовності апаратно-програмного комплексу	48
3.7	3D сканування за допомогою використання мапи глибин	49
3.8	Розробка програмного забезпечення	58
	Висновки до розділу 3	64
	ВИСНОВКИ	65
	ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	66
	ДОДАТОК А КОНЦЕПТУАЛЬНА СХЕМА АПАРАТНО-ПРОГРАМНОГО КОМПЛЕКСУ	69
	ДОДАТОК Б СТРУКТУРНА ДІАГРАМА АПАРАТНО-ПРОГРАМНОГО КОМПЛЕКСУ	70
	ДОДАТОК В ДІАГРАМА ПОСЛІДОВНОСТІ АПАРАТНО-ПРОГРАМНОГО КОМПЛЕКСУ	71
	ДОДАТОК Г ПРОГРАМНИЙ КОД АПАРАТНО-ПРОГРАМНОГО КОМПЛЕКСУ	72

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

GPIO	–	General-purpose input/output
HTTP	–	HyperText Transfer Protocol
IDE	–	Integrated development environment
LiDAR	–	Light Detection and Ranging
PCB	–	Printed circuit board
SLAM	–	Simultaneous localization and mapping
ToF	–	Time-of-flight

ВСТУП

В останні роки технології віртуальної та доповненої реальності набули високого розвитку та популярності і широко використовуються у різноманітних сферах від розважальної до наукової та інженерної. Застосування доповненої реальності надає широкі можливості в розвитку таких сервісів як навігація та орієнтування у тривимірному просторі, побудови динамічних та якісних моделей об'єктів у 3D-просторі, що дозволяє значно покращити картографування та аналіз певних ділянок земної поверхні.

Однак розробка такого роду сервісів вимагає значних матеріальних та людських ресурсів. Також деталізація тривимірних карт та моделей об'єктів з доповненими атрибутами вимагає чималих обчислювальних ресурсів, що не дозволяє використовувати такі сервіси широкому загалу споживачів та обмежує їх десктопними системами.

Актуальність кваліфікаційної роботи полягає в розробці та модифікації алгоритмів доповненої реальності з метою зменшення потрібних обчислювальних ресурсів для візуалізації 3D об'єктів на мобільних пристроях.

Мета: розробка AR-системи орієнтування в просторі на базі гіроскопічного датчика MPU-6050 та ToF-камери.

Об'єкт: процес координації руху роботизованих систем через відслідковування положення у просторі.

Предмет: методи та засоби отримання інформації про положення та відстань до об'єктів.

Завдання до проекту:

- провести аналіз апаратної частини та підібрати необхідні компоненти для повноцінної реалізації робочого дистанційного пристрою;
- створити схему підключення всіх елементів апаратної частини майбутнього проекту у програмному застосунку Fritzing;

- розробити програмне забезпечення для роботи та керування пристроєм;

- розробити алгоритм обробки отриманих даних.

Методи дослідження: В даній роботі будуть використовуватися як теоретичні та практичні методи дослідження. Із теоретичних будуть застосовані такі методи як функціональний, системний та моделювання. А із практичних методів будуть використані експериментальний та математичний методи.

За допомогою функціонального методу будуть досліджуватися принципи роботи гіроскопічних та акселерометричних датчиків, те як працюють ToF камери, та функції мікроконтролерів.

Систематичний метод дослідження покаже як працюють та взаємодіють між собою всі компоненти апаратної частини.

Експериментальним методом буде досліджено працездатність системи на практиці.

Гіпотеза дослідження полягає у тому чи використання датчика MPU-6050 та ToF камери дасть кращий результат, а ніж альтернативні рішення. Також чи покращаться навігаційні навички людини при використанні даної системи.

Новизна роботи полягає у розробці апаратно-програмного комплексу на базі MPU-6050 та ToF-камери для побудови віртуального середовища певної місцевості з елементами доповненої реальності у режимі реального часу віртуальній для покращення навігації.

Дане інтерактивне обладнання може бути використано у закладах освіти, музеях, інклюзивних центрах, будівництві, ріелторському бізнесі, для виконання роботи у небезпечних середовищах.

Робота пройшла апробацію під час XXV Всеукраїнської науково-практичної конференції «Могилянські читання» (Миколаїв, 07–11 листопада 2022 р.).

Публікації. Основні положення та результати магістерської роботи опубліковані у збірнику матеріалів XXV Всеукраїнської науково-практичної конференції «Могилянські читання–2022»[31].

1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ СИСТЕМИ, ЩО РОЗРОБЛЯЄТЬСЯ. ФОРМУВАННЯ ВИМОГ ДО АПАРАТНО-ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Методи орієнтації у просторі

Вміння орієнтації у просторі в історії людства завжди було нагальною проблемою для рішення якої були розроблені і досі розробляються нові методи, що дозволяють швидко та ефективно знаходити власне місцезнаходження та прокладання прокладати найкоротші маршрути. У кожного існуючого методу є своє переваги та недоліки тому як правило використовують комбінацію декількох способів навігації.

1.1.1. Ехолокаційний метод

Ехолокація – це спосіб, використовуючи який можна визначити положення об'єкта за часом затримки повернення відбитої хвилі. Ехолокація може бути заснована на відбитті ультразвуку, звуку, радіохвиль, а також сигналів випромінювання з різними частотами. Радіолокація використовується, якщо хвилі є радіохвилями, в іншому випадку - звукова локація. Локація може відбуватися в газоподібному, рідкому або твердому середовищі і може бути як активною, так і пасивною [1].

При активній акустичній локації виробляється звукова хвиля для створення відлуння, яке потім досліджується для встановлення місцезнаходження конкретного об'єкта.

Процес пасивної акустичної локації передбачає виявлення звуку або вібрації, які виробляє об'єкт, що потім виявляється і аналізується для точного визначення місцезнаходження об'єкта.

При дифузному розміщенні датчиків (рис. 1.1) перетворювач виконує функції як випромінювача, так і приймача і, як правило, розміщується в одному корпусі з електронним перетворювачем, а при протилежному розміщенні датчиків (рис. 1.2) датчик-випромінювач розміщується навпроти датчика-приймача. Датчик або визначає час проходження хвилі від нього до об'єкта і назад (дифузний режим роботи), або визначає, чи прийняв інший приймач сигнал, який був переданий (опозитний режим роботи).

Датчики дифузного типу являється найбільш розповсюдженим. У них перетворювач служить як випромінювачем, так і приймачем, і зазвичай знаходиться в одному корпусі з електронним перетворювачем.. Це мінімізує вартість монтажу, так як необхідно монтувати і підключити тільки один прилад.

Найпоширенішим видом датчиків є дифузійний тип. Перетворювач в них виконує функції як випромінювача, так і приймача, і часто розміщується в одному корпусі з електронним перетворювачем. Завдяки тому, що потрібно розмістити і з'єднати лише один пристрій, зменшуються витрати на установку.

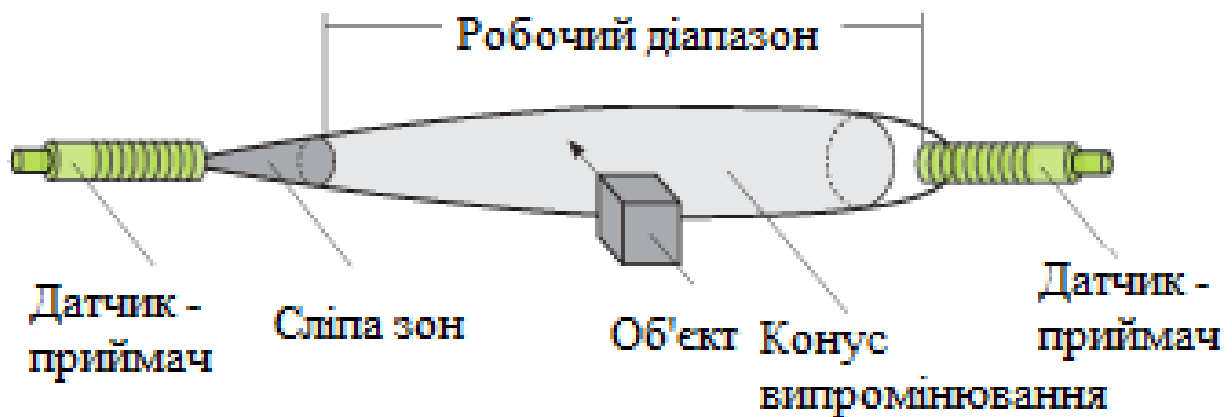


Рисунок 1.1 – датчики дифузного типу

В датчиках опозитного типу передавач і приймач є окремими пристроями і монтуються один навпроти одного. Вихід вимикача активізується, якщо ультразвуковий пучок переривається об'єктив ці датчики не мають потреби в постійному перемиканні між режимами передачі і

Кафедра комп'ютерної інженерії
Апаратно-програмний комплекс для real-time симуляції руху об'єкту на базі
гіроскопічного датчика MPU-6050
прийому і не очікують надходження відбитого сигналу. Завдяки цьому вони
мають значно більш швидкий час відгуку і високу частоту перемикання.

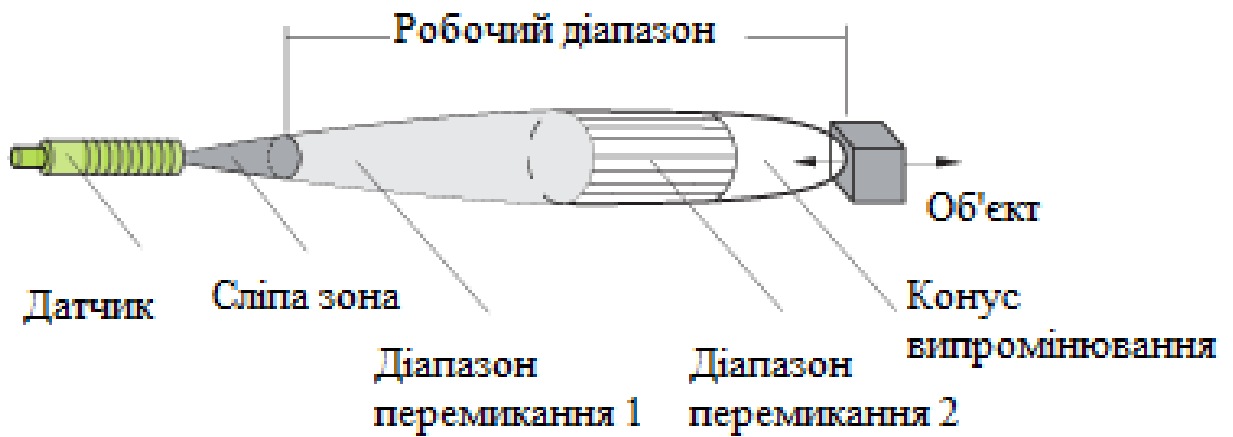


Рисунок 1.2 – датчики опозитног типу

0.1.2 SLAM

SLAM (англ. simultaneous localization and mapping) - це технологічний метод картографування, який дозволяє роботам та іншим автономним транспортним засобам будувати мапу невідомого оточення і одночасно локалізувати себе на цій карті. Використовуючи широкий спектр алгоритмів, обчислень та інших сенсорних даних, програмні системи SLAM дозволяють роботам або іншим транспортним засобам, таким як безпілотні літальні апарати або самокеровані автомобілі, прокладати курс у незнайомому середовищі, одночасно визначаючи власне місцезнаходження в цьому середовищі [2].

Такий підхід до самолокалізації дозволяє наносити на карту райони, які можуть бути занадто малими або занадто небезпечними для дослідження людиною.

Технологія одночасної локалізації і картографування вже широко використовується в усьому світі - від домашніх роботів-пилососів до автомобілів.

По мірі того, як ця технологія стає дешевшою і все більше досліджень проводиться на цю тему, з'являється низка нових практичних прикладів використання SLAM в широкому спектрі галузей промисловості.

Всі рішення SLAM включають в себе якийсь пристрій або інструмент, який дозволяє роботу або іншому транспортному засобу спостерігати і вимірювати навколишнє середовище навколо нього.

Це можуть робити камери, інші типи датчиків зображення, технологія лазерного сканера LiDAR і навіть гідролокатор. По суті, будь-який пристрій, який може бути використаний для вимірювання фізичних властивостей, таких як місцезнаходження, відстань або швидкість, може бути включений як частина системи SLAM.

Після того, як ці вимірювання розраховані, система SLAM повинна мати певне програмне забезпечення, яке допомагає інтерпретувати ці дані. Існує широкий спектр варіантів, доступних на цьому фронті, починаючи від серії алгоритмів переплетення і закінчуючи іншими типами складного узгодження сканування.

Всі ці "внутрішні" рішення, по суті, слугують одній меті: вони витягують сенсорні дані, зібрані пристроєм вимірювання дальності, і використовують їх для ідентифікації орієнтирів у невідомому середовищі.

Належним чином функціонуюче рішення SLAM передбачає постійну взаємодію між пристроєм вимірювання дальності, програмним забезпеченням для вилучення даних, самим роботом або транспортним засобом і додатковим обладнанням, програмним забезпеченням або іншими технологіями обробки даних [3].

Усі ці елементи можуть змінюватися в залежності від випадку використання, але для того, щоб будь-яка система SLAM могла точно досліджувати навколишнє середовище, усі ці елементи повинні безперервно працювати разом.

0.1.3 LiDAR

LiDAR (англ. Light Detection and Ranging) технологія вимірювання відстаней шляхом випромінювання світла (лазер) і виміру часу повернення цього відбитого світла на приймач. В основному, це метод використання лазерів для збору даних про поверхню, які потім можуть бути перероблені в 3D-моделі. Технологія LiDAR є відносно новою. Вперше вона була розроблена на початку 1960-х років, коли інженери та землевпорядники почали використовувати її для створення карт річок, струмків та інших географічних об'єктів.

LiDAR помітно відрізняється від інших систем тривимірного моделювання, таких як фотограмметрія, яка передбачає отримання сотень або навіть тисяч окремих фотографій і зшивання їх в одну детальну модель [4]. Натомість LiDAR використовує світлову енергію і складні лазери для сканування і вимірювання цілі на відстані. Прикріплений до високолітаючого безпілотної, LiDAR може прорізати або обходити листя та інше сміття так, як це не можуть зробити фотограмметричні системи, що робить його потужним інструментом для відкриття нових об'єктів в такій галузі, як археологія.

Процес схожий на те, як сонар або радар виявляють об'єкти (рис.1.3), які покладаються на відбиття звукової або радіохвилі від об'єкта і вимірювання часу, необхідного для її повернення, щоб зрозуміти, наскільки далеко знаходиться об'єкт або які його розміри.

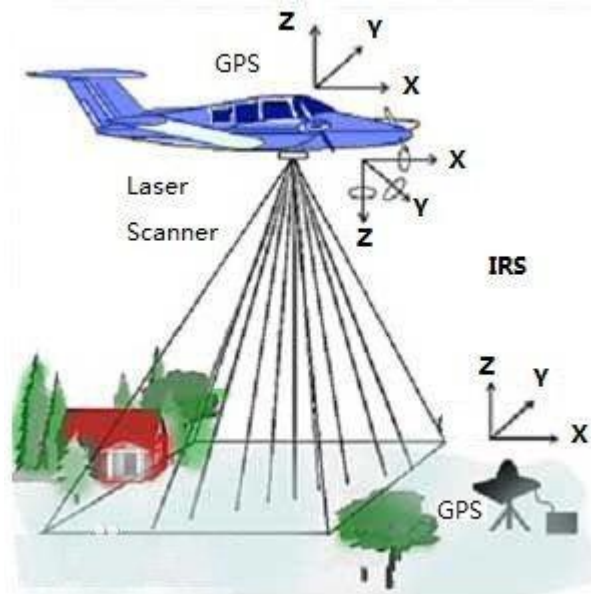


Рисунок 1.4 – Принцип роботи LiDAR

За допомогою LiDAR потужний лазер стріляє точними імпульсами в ціль і вимірює імпульс, який відбивається назад, для того, щоб зібрати дані про ціль.

Щоб створити уявлення про об'єкт у трьох вимірах, датчики LiDAR проводять вимірювання:

- Час - скільки часу потрібно для повернення імпульсу;
- Інтенсивність - силу зворотного лазерного імпульсу;
- Кут відбиття - як змінюється вимірювана поверхня, на що вказує кут відбиття.

Після збору цих даних спеціалізоване програмне забезпечення для 3D-картографування обробляє цю інформацію разом з даними GPS (Глобальної системи позиціонування) та INS (Інерціальної системи навігації) для створення детальних і точних 3D-моделей цільової області або об'єкта [5].

0.1.4 ToF Camera

ToF Camera (англ. Time-of-flight camera) - технологія яка пропонує 3D-зображення з використанням активного модульованого джерела світла і

Кафедра комп'ютерної інженерії
Апаратно-програмний комплекс для real-time симуляції руху об'єкту на базі гіроскопічного датчика MPU-6050 недорогої CMOS-матриці, робить революцію в індустрії машинного зору. TOF-камери є привабливим вибором для різноманітних застосувань завдяки своїм невеликим розмірам, простоті використання, відмінній точності та частоті кадрів. Потім розглядаються численні застосування TOF-сенсорів, включаючи жестикуляцію і 3D-сканування і 3D-друк [6].

PA
GE
*
M
ER
GE
FO
R
^^

Модульоване джерело світла використовується для освітлення сцени в 3D-камері часового прольоту (TOF), яка потім дивиться на відбите світло. Відстань розраховується за фазовим зсувом між освітленням і відбиттям. Рис. 1.4 демонструє фундаментальну ідею TOF.

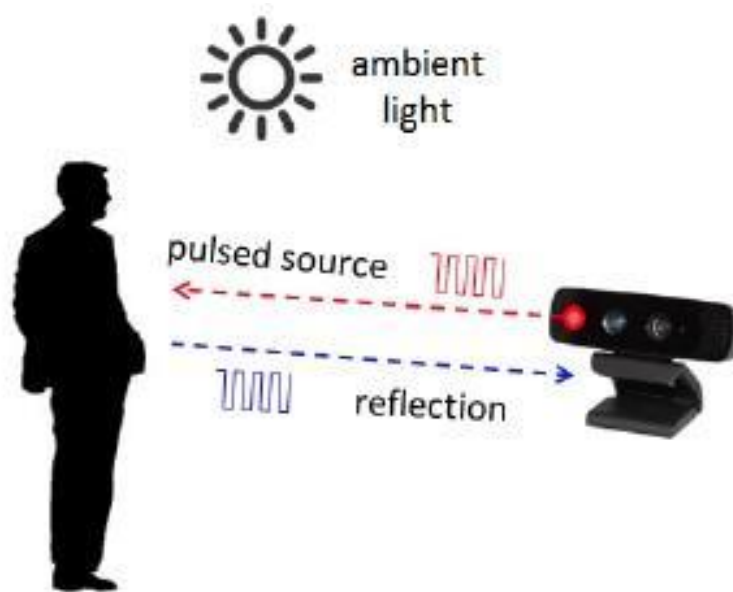


Рисунок 1.4 – Робота ToF камери

Освітлення зазвичай забезпечує твердотільний лазер або світлодіод, який випромінює світло в ближньому інфрачервоному спектрі (850 нм), невидимому для людського ока. Світло вловлюється датчиком зображення, побудованим так, щоб реагувати на той самий спектр, який перетворює фотонну енергію в електричний струм. Слід пам'ятати, що світло, яке потрапляє на сенсор, складається з двох частин: навколишнього та відбитого. Тільки відбита складова має інформацію про відстань (глибину), що міститься в ній. Висока навколишня складова знижує відношення сигнал/шум в результаті [7].

Джерело світла імппульсується або модулюється джерелом безперервної хвилі, зазвичай синусоїдою або прямокутною хвилею, для того, щоб виявити зміни фази між освітленням і відбиттям. Модуляція прямокутної хвилі є більш поширеною, оскільки цифрові схеми роблять її простою в реалізації.

Інтегруючи фотоелектрони з відбитого світла або запускаючи швидкий лічильник при першій ознаці відбиття, можна здійснити імппульсну модуляцію. Для останнього необхідний швидкий фотоприймач, часто однофотонний лавинний діод. Цей метод підрахунку вимагає швидкої електроніки, тому що потрібно 6,6 пікосекунд, щоб визначити час імппульсу з точністю до 1 міліметра. При кімнатній температурі отримати такий рівень точності в кремнії практично неможливо.

Світлові імппульси або один світловий імппульс використовуються в найпростішій хронометражній камері. Короткий проміжок часу витрачається на включення освітлення; отриманий світловий імппульс освітлює сцену і відбивається від розташованих поруч об'єктів. Відбите світло збирається об'єктивом камери і потрапляє на фокальну площину матриці або сенсора.

У найпростішому варіанті часової камери використовуються світлові імппульси або одиночний світловий імппульс (рис. 1.5). Підсвічування включається на дуже короткий час, отриманий світловий імппульс висвітлює сцену і відбивається від об'єктів, що знаходяться в полі зору. Об'єктив камери збирає відбите світло і формує зображення на матриці або матриці фокальної площини[8].

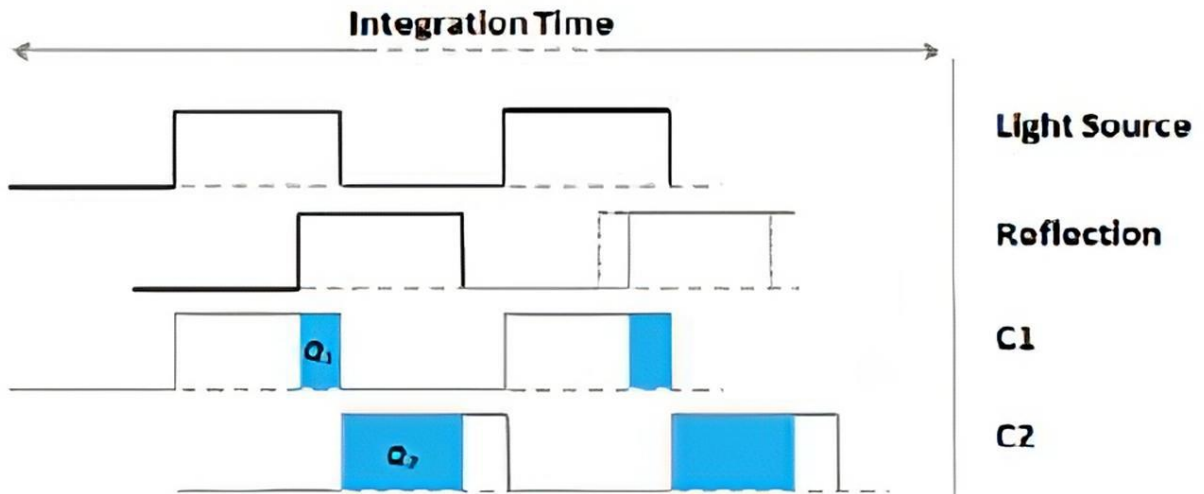


Рисунок 1.5 – Метод одиночного імпульсу

Імпульсний метод простий. Джерело світла світить протягом короткого періоду часу Δt , а відбита енергія відбирається в кожному пікселі паралельно за допомогою двох синфазних вікон, C1 і C2, з однаковою тривалістю Δt . Електричні заряди, накопичені під час цих вибірок, Q_1 і Q_2 , вимірюються і використовуються для обчислення відстані за рівнянням (1.1):

$$d = \frac{1}{2} C \Delta t \left(\frac{Q_1}{Q_1 + Q_2} \right) \quad (1.1)$$

де C – швидкість світла, Δt – тривалість імпульсу, Q_1 – накопичений заряд в пікселі при випромінюванні світла, Q_2 – накопичений заряд при його відсутності.

На відміну першого методу, метод безперервної хвилі бере кілька зразків для кожного вимірювання, причому кожен зразок зсувається по фазі на 90 градусів, загалом чотири зразки (рис. 1.6). Проводиться вимірювання декількох відносних фаз між тактовим сигналом пікселя і сигналом лазерної модуляції. Ці вимірювання об'єднуються для визначення фази основної у зворотному модульованому світловому сигналі. Знання цієї фази дозволяє розрахувати час, за який світло проходить шлях від джерела світла до об'єкта спостереження і назад до пікселя сенсора.

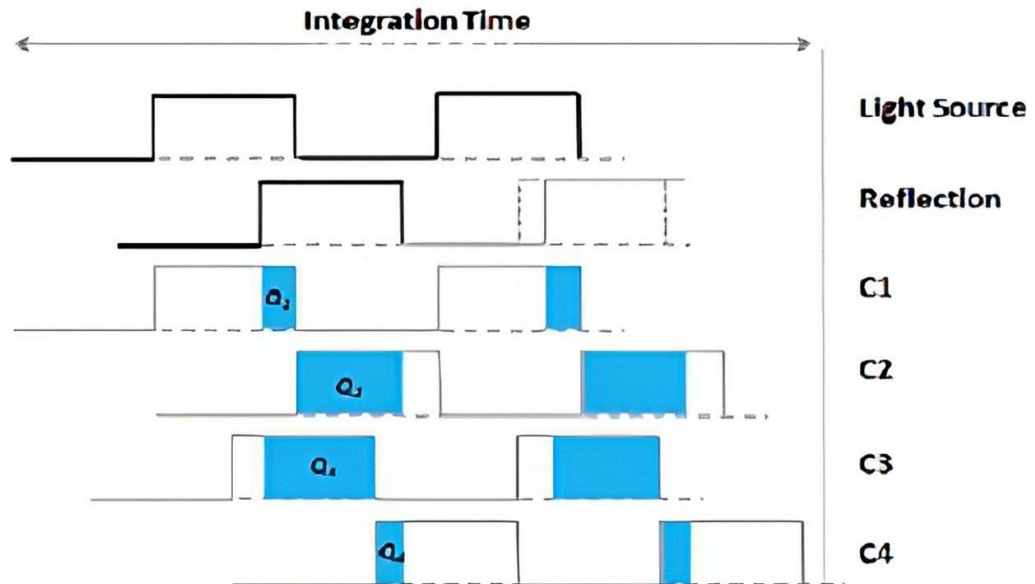


Рисунок 1.6 – Метод безперервної хвилі

Використовуючи цю методику, фазовий кут між освітленням і відбиттям, відстаню d , можна розрахувати за рівнянням (1.2):

$$d = \frac{ct}{2\pi} \arctan \arctan \frac{Q_3 - Q_4}{Q_1 - Q_2} \quad (1.2)$$

На перший погляд, складність методу безперервної хвилі, у порівнянні з імпульсним методом, може здатися невиправданою, але більш уважний погляд на рівняння КС показує, що члени $(Q_3 - Q_4)$ і $(Q_1 - Q_2)$ зменшують вплив постійного зсуву від вимірювань. Крім того, коефіцієнт у фазовому рівнянні зменшує вплив постійних підсилень від вимірювань відстані, таких як підсилення і ослаблення системи, або відбитої інтенсивності. Це бажані властивості.

0.1.5 Інерціальна навігація

Інерціальна навігаційна система - це автономний пристрій, що складається з інерціального вимірювального пристрій та обчислювального блоку. інерціального вимірювального пристрій, як правило, складається з 3-

осьового акселерометра, 3-осьового гіроскопа та іноді 3-осьового магнітометра і вимірює кутову швидкість та прискорення системи. Обчислювальний блок використовується для визначення орієнтації, положення і швидкості системи на основі необроблених вимірювань, отриманих від інерціального вимірювального блоку, враховуючи початкове стартове положення і орієнтацію. Точність інерціальної навігаційної системи (ІНС) визначається якістю її інерціального вимірювального пристрою (ІВП), який зазвичай складається з трьохосьового акселерометра, трьохосьового гіроскопа і, в деяких випадках, магнітометра. Високоточні INS можуть досягати точності визначення місцезнаходження до 2 см і швидкості до 0,05 км/год. INS у поєднанні з доплерівським реєстратором швидкості (DVL) може забезпечити точність навігації краще, ніж 0,1% від пройденої відстані, що означає, що транспортний засіб, який проїхав 100 км, знатиме своє місцезнаходження з точністю понад 100 м наприкінці пробігу (рис. 1.7).

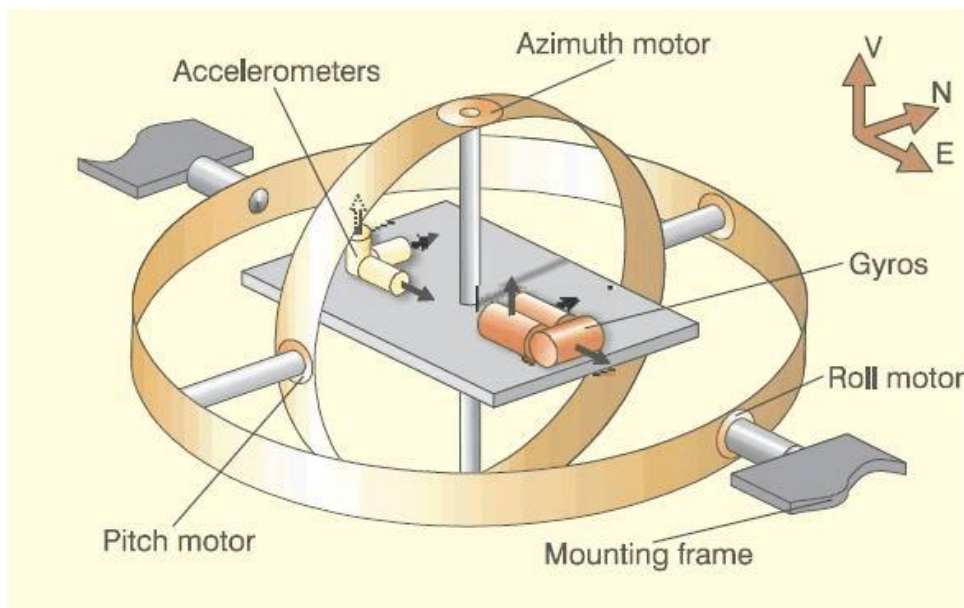


Рисунок 1.7 – Будова інерціальної навігаційної системи

Часто інерціальні датчики доповнюються барометричним висотоміром, а іноді магнітними датчиками (магнітометрами) та/або пристроями для вимірювання швидкості. Інерціальна навігаційна система використовуються на мобільних роботах і на таких транспортних засобах, як кораблі, літаки,

підводні човни, керовані ракети і космічні апарати. Інші терміни, що використовуються для позначення інерціальних навігаційних систем або тісно пов'язаних з ними пристроїв, включають інерціальну систему наведення, інерціальний прилад, інерціальний вимірювальний пристрій і багато інших варіацій [9].

Положення системи розраховується шляхом інтегрування кутової швидкості (кутового переміщення), вимірюваної гіроскопами за певний проміжок часу. Для цілей цього аналізу ми розглядаємо одновісний рух, оскільки нелінійний зв'язок положення у загальному випадку унеможливорює багатовісний аналіз. Рівняння для вимірюваної кутової швидкості для однієї осі може бути представлено з урахуванням джерел похибок за наступним рівнянням (1.3):

$$w = (1 + k)\omega_t + b_g + n_g \quad (1.3)$$

де w – вимірювана кутова швидкість, ω_t – істинна кутова швидкість, k – похибка масштабного коефіцієнта, b_g – зсув, що змінюється в часі, і n_g – випадковий гаусівський шум.

Безпілотна інерціальна навігація, також відома як "мертвий розрахунок", вимагає надзвичайно точних інерціальних датчиків для забезпечення достатньо точної оцінки положення і швидкості для навігаційних цілей. Різноманітні джерела помилок в самих вимірюваннях інерціальних датчиків призводять до необмеженого зростання помилок в навігаційному рішенні інерціальна навігаційна система, таких як зсув, шум, помилки масштабного фактору, неспіввісність, температурні залежності і g-чутливість гіроскопа. Характеристики гіроскопа, як правило, домінують над

Кафедра комп'ютерної інженерії
Апаратно-програмний комплекс для real-time симуляції руху об'єкту на базі
гіроскопічного датчика MPU-6050
похибками позиціонування при виконанні чистої інтеграції інерціальних
датчиків [10].

Акселерометри в системі інерціальна навігаційна система вимірюють як лінійне прискорення системи, обумовлене рухом, так і псевдоприскорення, викликане силою тяжіння. Для отримання лінійного прискорення системи, зумовленого рухом, псевдоприскорення, спричинене силою тяжіння, необхідно відняти від вимірювань акселерометра з використанням оцінок положення системи. Отримане значення лінійного прискорення можна інтегрувати один раз, щоб отримати швидкість системи, і двічі, щоб отримати положення системи. Проте, ці обчислення сильно залежать від підтримки інерціальної навігаційної системаиточної оцінки орієнтації, оскільки будь-яка помилка в орієнтації викликає помилку в обчисленому прискоренні, що, в свою чергу, викликає помилки в інтегрованому положенні і швидкості.

0.2 Вимоги до системи

Для побудови надійної і точної системи навігації необхідно забезпечити баланс між з такими критеріями як вартість, енергоефективність, простота калібрування, складністю розгортання і точністю.

Пристрій має бути компактним для переустаткування на різні мобільні системи як дрони. Система повина працювати з невеликим енергоспоживанням. Собівартість має бути компромісом між якістю та доступністю.

Під час орієнтації шум може створюватися змінами навколишнього середовища, а також ефектами, викликаними джерелами світлового шуму і перешкод, такими як відбиті ехосигнали, перешкоди від об'єктів або

дифракція сигналу. Ці завади повинні бути досліджені для того, щоб відфільтрувати шум вимірювання і підвищити точність оцінки місцезнаходження.

0.3 Висновки до розділу

В даному розділі було проаналізовано низку існуючих методів навігації у просторі та поставлені вимоги до системи. В результаті було обрано рішення створення гнучкої, економічної і мало енерговитратної системи. Для навігації використовується інерційна система навігації, для доповнення недоліків у комбінації використовується ToF Camera для динамічного створення оточення у 3-вимірному просторі.

2. АПАРАТНО-ПРОГРАМНИХ КОМПОНЕНТІВ

1.1 Огляд компонентів апаратно-програмного комплексу

Після проведення аналізу проблемної області та перед початком проектування апаратно-програмного комплексу необхідно провести аналіз існуючої апаратної частини, а саме мікроконтролерів та датчиків. На базі зібраних даних необхідно обрати компоненти які будуть використанні для реалізації системи здатної в режимі реального часу створювати 3D моделі.

1.1.1 Вибір мікроконтролера

Arduino - це електронна платформа з відкритим вихідним кодом, що базується на принципі простого використання апаратного та програмного забезпечення. Arduino невелика плата управління з власним процесором і пам'яттю. Крім того, на платі є кілька десятків контактів, до яких можна приєднувати різні компоненти. Плати Arduino можуть зчитувати вхідні дані, такі як світло на датчику, палець на кнопці або твіт у Twitter, і перетворювати їх у вихідні дані, такі як запуск двигуна, увімкнення світлодіода або публікація чогось в Інтернеті. Протягом багатьох років Arduino слугувала мозковим центром сотень проектів, починаючи від простих побутових пристроїв і закінчуючи складними науковими приладами. Глобальна спільнота розробників - студентів, аматорів, художників, програмістів і професіоналів - зібралася навколо цієї платформи з відкритим вихідним кодом, і їхній внесок створив вражаючу кількість доступних знань, які можуть бути надзвичайно корисними як для початківців, так і для експертів [11].

Arduino використовується в мільйонах різних проектів і додатків завдяки простому і доступному користувацькому інтерфейсу. Програмне забезпечення Arduino просте у використанні для початківців і водночас достатньо гнучке для досвідчених користувачів. Воно сумісне з Mac, Windows та Linux. Викладачі та студенти використовують його для

створення недорогого наукового обладнання, демонстрації принципів хімії та фізики, а також для початку вивчення програмування та робототехніки. Дизайнери та архітектори використовують його для створення інтерактивних прототипів, а музиканти та художники - для інсталяцій та тестування нових музичних інструментів.

Arduino Nano (рис. 2.1) - це плата мікроконтролера на базі ATmega328P, або Atmega168. Вона дуже схожа на плату Arduino Uno, проте з точки зору розташування виводів і функцій ця плата обійшла Arduino Uno завдяки своїм компактним розмірам. Як ми всі знаємо, при створенні вбудованої системи бажано використовувати компоненти компактного розміру. Плати Arduino в основному використовуються для створення електронних проектів, таких як вбудовані системи, роботи тощо. Однак наноплати в першу чергу призначені для початківців з невеликим технічним досвідом. Він містить все необхідне для підтримки мікроконтролера, щоб почати роботу достатньо підключити його до комп'ютера через USB або заживити його за допомогою перетворювача змінного/постійного струму або батареї [12].

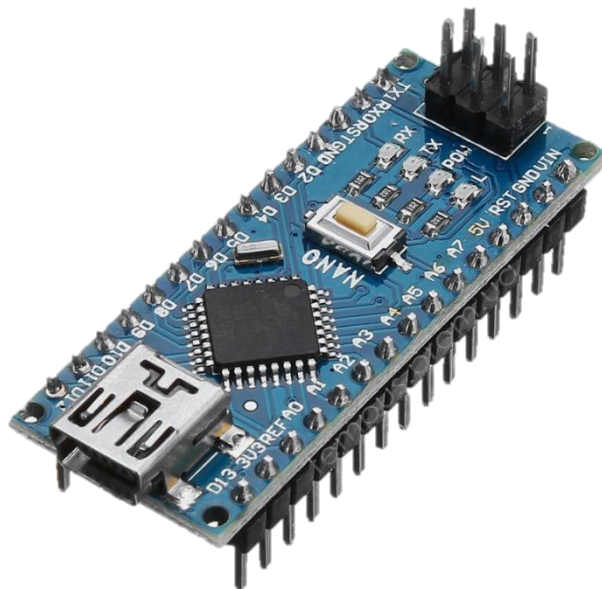


Рисунок 2.1 – Зовнішній вигляд плати Arduino Nano

З моменту своєї появи Arduino Nano стала великим хітом серед ентузіастів електроніки - від початківців до досвідчених програмістів. Це платформа з відкритим вихідним кодом, що означає, що плати та програмне забезпечення знаходяться у вільному доступі, і будь-хто може змінювати та оптимізувати їх для покращення функціональності. Програмне забезпечення, що використовується для пристроїв Arduino, відоме як IDE), і воно є безкоштовним у використанні, але вимагає деяких базових навичок для розуміння. Програмувати можна мовами C та C++.

Зв'язок з платою Arduino Nano може здійснюватися за допомогою різних засобів, включаючи додаткову плату Arduino, комп'ютер або мікроконтролери. Мікроконтролер, що використовується в Nano платі (ATmega328), забезпечує послідовний зв'язок (UART TTL). Доступ до нього можна отримати через цифрові виводи, такі як TX та RX. Програмне забезпечення Arduino має контролер послідовного з'єднання, який дозволяє легко передавати та отримувати текстові дані від пристрою.

Щоразу коли інформація надсилається через FTDI і USB в напрямку комп'ютера, світлодіоди TX і RX на платі Nano блимають. Бібліотека SoftwareSerial дозволяє здійснювати послідовний зв'язок з будь-яким цифровим виводом плати. Мікроконтролер також підтримує зв'язок SPI і I2C (TWI).

Програмування Arduino nano можна виконати за допомогою програмного забезпечення Arduino. Виберіть наноплату в меню Tools (Інструменти). Мікроконтролер ATmega328 на нано-платі поставляється з попередньо запрограмованим завантажувачем. Цей завантажувач дозволяє завантажувати новий код без необхідності використання зовнішнього апаратного програматора. Це можна зробити за допомогою протоколу STK500. Тут також можна обійти завантажувач, і програмувати мікроконтролер за допомогою заголовка внутрішньосхемного послідовного програмування (ICSP) з провайдером Arduino ISP.

Nano сконструйовано таким чином, що перезавантаження виконується програмним забезпеченням, а не натисканням кнопки на платформі перед додаванням свіжого коду. Конденсатор ємністю 100 нФ з'єднує одну з ліній керування потоком даних (DTR) FT232RL з виводом перезавантаження ATmega168 або ATmega328. Активація цієї лінії, тобто подача сигналу низького рівня, призводить до перезавантаження мікроконтролера. Використовуючи цю функцію, програма Arduino завантажує код одним натисканням кнопки Upload в середовищі програмування. Сигнал низького рівня на лінії DTR синхронізується з початком написання коду, зменшуючи час очікування завантажувача [13].

На платі встановлено мікроконтролер Atmega328 з великою кількістю функцій, таких як таймери, лічильники, переривання, ШІМ, процесор, виводи вводу/виводу, а також тактова частота 16 МГц, яка допомагає виробляти більшу частоту і кількість інструкцій/цикл. У таблиці 2.1 наведено детальну характеристику плати Arduino Nano.

Таблиця 2.1 – Характеристики плати Arduino Nano

Мікроконтролер	ATmega328P
Тип корпусу	TQFP-32
Вхідна напруга	7-12 В
Робоча напруга	5 В
Цифрових входів / виходів	14
Аналогових входів	8
Сила струму на входах / виходах	40 мА
Сила струму для 3.3В виходу	50 мА
Пам'ять	32 кБ
Тактова частота	16 МГц
Розмір	18 x 45 мм

Raspberry Pi - це назва серії одноплатних комп'ютерів, створених Raspberry Pi Foundation, британською благодійною організацією, метою якої є навчання людей комп'ютерним технологіям та полегшення доступу до комп'ютерної освіти.

Люди по всьому світу використовують Raspberry Pi для розвитку навичок програмування, створення апаратних проектів, автоматизації своїх будинків, використання периферійних обчислень і кластерів Kubernetes і навіть для промислового застосування.

Окрім того, що Raspberry Pi є відносно доступним комп'ютером під управлінням Linux, він також пропонує набір виводів GPIO (загального призначення), які дозволяють досліджувати Інтернет речей та керувати електронними компонентами для фізичних обчислень[14].

Raspberry Pi 4 Model B (рис. 2.2) - перший комп'ютер четвертої ітерації Raspberry Pi. Нова модель включає більш потужний чотирьохядерний ARM-процесор Broadcom BCM2711B0 і відеопроцесор Broadcom VideoCore VI з підтримкою 4K, а також більш швидкісні з'єднання USB 3.0 і USB Type-C для живлення.



Рисунок 2.2 – Зовнішній вигляд плати Raspberry Pi 4 Model B

Він випускається в трьох варіантах, єдиною відмінністю яких є об'єм оперативної пам'яті LPDDR4 SDRAM, встановленої на платі. Базова модель має 1 ГБ оперативної пам'яті. Розширенні моделі мають збільшений обсяг оперативної пам'яті розміром 2 ГБ , 4 ГБ та 8 ГБ.

Лінійка недорогих комп'ютерів Raspberry Pi відома тим, що вони не більші за колоду карт. Raspberry Pi 4 Model B має розміри всього 86 x 56 x 10 мм. Він також майже не відрізняється за вагою - 46 грамів, що лише на грам важче, ніж попередні версії, такі як Raspberry Pi 3 B+ 45 грамів.

Перелік портів на Raspberry Pi 4 також нагадує про фізичні можливості маленького комп'ютера: чотири порти USB (два USB 3.0 і два USB 2.0), пара портів mini HDMI 2.0 і роз'єм живлення USB Type-C. 3,5-міліметровий роз'єм слугує аналоговим аудіо/відео портом, а Gigabit Ethernet - з повною пропускною здатністю, яка була недоступна в попередніх версіях Pi - забезпечує дротовий зв'язок, а також живлення, завдяки підтримці Power over Ethernet. Для бездротового зв'язку Pi має Wi-Fi 802.11ac та Bluetooth 5.0[15].

Оскільки нова SoC вимагає більшої потужності, Raspberry Pi 4 B заряджається через USB Type-C, а не через мікро USB. Для цього також потрібен адаптер живлення, здатний видавати щонайменше 3 ампера і 5 вольт, хоча можна обійтися і 2,5 амперами, якщо ви не підключаєте багато периферійних пристроїв до USB-портів. Окрім вимог до живлення, роз'єми USB Type-C є реверсивними, що значно полегшує їх підключення.

Raspberry Pi 4 має схожий дизайн і розміри зі своїми попередниками, але це абсолютно нова платформа, яка працює на новому процесорі Broadcom BCM2711B0. З моменту випуску першого Pi в 2012 році всі Pi використовували 40-нм SoC, але цей новий чіп заснований на 28-нм техпроцесі, і замість старої мікроархітектури Cortex-A53 він використовує Cortex-A72. BCM2711B0 в Raspberry Pi 4 має чотири процесорних ядра і тактову частоту 1,5 ГГц, що, на перший погляд, не набагато швидше, ніж чотирьохядерний 1,4-гігагерцовий BCM2837B0 в Raspberry Pi 3B+.

Однак, Cortex A72 має глибину конвеєра в 15 інструкцій, порівняно з глибиною конвеєра попередньої моделі в 8 інструкцій, а також підтримує позачергове виконання, тобто не чекає на завершення однієї операції, щоб почати іншу. Навіть при однаковій тактовій частоті (хоча BCM2711B0 побудований на нижчому виробничому вузлі), процесори Cortex-A72 будуть значно швидшими і споживатимуть більше енергії, ніж попередники, що працюють на базі A53. Характеристики плати Raspberry Pi 4 Model B наведені у табл. 2.3.

Таблиця 2.3 – Характеристики плати Raspberry Pi 4 Model B

SoC	ARM Cortex-A72 Quad Core
Тактова частота	1.5 МГц
Кількість ядер процесора	4
RAM	1ГБ/2ГБ/4ГБ/8ГБ
GPU	VideoCore VI
Вхідна напруга	96 - 264 В
Робоча напруга	5 В
USB	2 x USB 2.0, 2 x USB 3.0
Цифрових входів / виходів	40
WiFi	2.4 МГц / 5 МГц
Розмір	88 x 58 мм

1.1.2 Вибір гіроскопічного датчика

MPU-6050 (рис. 2.3) - це шестикоординатний пристрій відстеження руху, розроблений для смартфонів, планшетів і датчиків, що носяться, які потребують низького енергоспоживання, низької вартості та хорошої продуктивності. На одному кремнієвому кристалі розміщено 3-осьовий гіроскоп і 3-осьовий акселерометр, а також вбудований цифровий процесор руху (DMP). MPU-6050 доступний у різних конфігураціях, включаючи

окремі плати, модулі та мікросхеми, і може бути використаний у таких проектах, як роботи, дрони, ігрові контролери тощо. Adafruit продає плату MPU-6050 6-DoF Acceleration and Gyro Sensor з допоміжною схемою, яка дозволяє клієнтам використовувати датчик з логічними рівнями 3,3 В або 5 В [16].

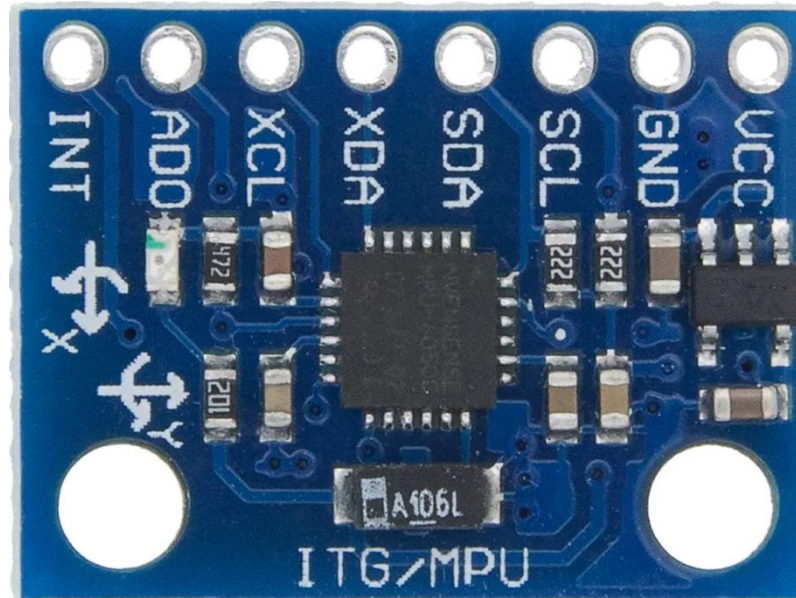


Рисунок 2.3 – Зовнішній вигляд гіроскопічного датчика MPU-6050

Цей модуль також має цифровий процесор руху (DMP), який достатньо потужний, щоб виконувати складні обчислення і таким чином звільняти час мікроконтролера. Модуль також містить два допоміжні виводи, які можна використовувати для підключення зовнішніх ІС-модулів, таких як магнітометр, але це необов'язково. Оскільки ІС-адреса модуля є програмованою, декілька датчиків MPU6050 можуть бути підключені до мікроконтролера через вивід AD0. Цей модуль також має добре задокументовані та оновлені бібліотеки, що робить його дуже простим у використанні з відомими платформами, такими як Arduino. Отже, якщо поставленна задача пошуку датчику для керування рухом радіокерованого автомобіля, дрона, самобалансуючого робота, гуманоїда, двоногого робота або чогось подібного, цей датчик буде чудовим рішенням[17].

Для точного відстеження як швидких, так і повільних рухів деталі оснащені програмованим користувачем повномасштабним гіроскопом з діапазоном ± 250 , ± 500 , ± 1000 і ± 2000 $^{\circ}/c$ (dps), а також програмованим користувачем повномасштабним акселерометром з діапазоном $\pm 2g$, $\pm 4g$, $\pm 8g$ і $\pm 16g$. Додаткові функції включають вбудований датчик температури і вбудований генератор з варіацією $\pm 1\%$ в робочому діапазоні температур. Більш детальна характеристики датчику MPU-6050 наведені у табл. 2.3

Напруга живлення	3.3 - 5.0 В
Діапазон гіроскопа	± 250 ± 500 ± 1000 ± 2000 $^{\circ}/c$
Діапазон акселерометра	± 2 ± 4 ± 8 ± 16 g
Протокол	ІІС (I2C / TWI)
АЦП	16-бітний
Розміри плати	20 x 16 мм

Таблиця 2.3 – Характеристики датчику HC-SR04

WT61 - це простий 3-осьовий акселерометр та 3-осьовий гіроскоп (рис. 2.4). Він має три діапазони детектування, що налаштовуються: $\pm 2g$, $4g$ і $8g$. Він має такі можливості, як визначення орієнтації, визначення одинарного та подвійного дотику та режими низького енергоспоживання. Оскільки це цифровий датчик, який взаємодіє через інтерфейс I2C, можна отримати надійні, чисті від шуму дані, взаємодіючи з акселерометром лише за допомогою двох виводів мікроконтролера [18].

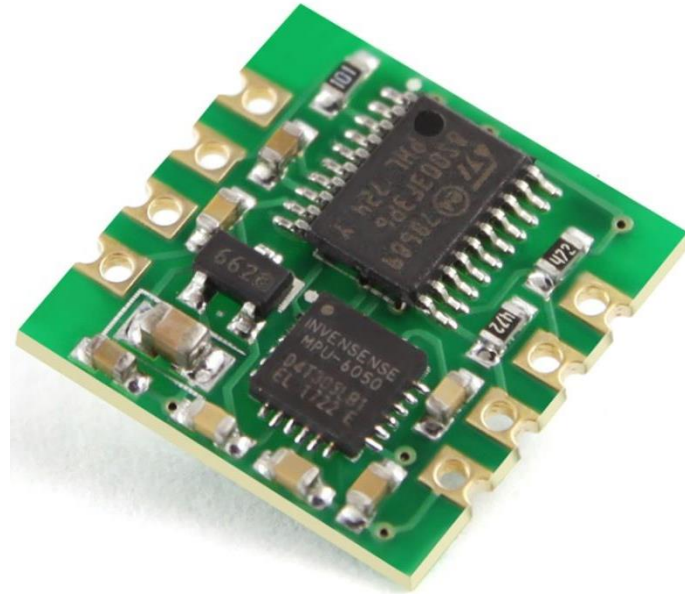


Рисунок 2.4 – Зовнішній вигляд датчика WT61

Найважливішим попередженням буде те, що WT61 має максимальний діапазон напруги 3.3 - 5 В, що стосується як джерела живлення, так і виводів I2C. Якщо під час роботи використовується датчик в системі з напругою 3.3 В, можливо просто підключити його; однак, якщо у проєкті він використовується з пристроєм з напругою 5 В (наприклад, Arduino Uno), необхідне деяке зміщення рівнів напруги між пристроями. Характеристики датчику WT61 наведені у табл. 2.4

Напруга живлення	3.3 - 5 В
Діапазон гіроскопа	$\pm 250 \pm 500 \pm 1000 \pm 2000$ °/с
Діапазон акселерометра	$\pm 2g / \pm 4g / \pm 6g$
Протокол	I2C
АЦП	16-бітний
Розміри плати	20 x 20 мм

Таблиця 2.4 – Характеристики датчику WT61

HMC5883L має індивідуальну адресу I2C, яка може бути корисною, якщо у проєкті використовується багато HMC5883L на одній шині, або, у випадку коли під час роботи виникає конфлікт адрес). Щоб вибрати адресу, підключіть вивід акселерометра під назвою "SA0" до живлення або заземлення [19].

За замовчуванням перемикач відкритий, що підтягує вивід SA0 до землі (на верхній стороні плати є резистор, який допомагає виконати це завдання). Якщо буде замкнено перемикач, застосувавши невелику краплю припою, щоб з'єднати обидві колодки разом, SA0 буде замкнений на землю.

1.1.3 Вибір ToF камери

ArduCam ToF Camera – невелика камера, що за допомогою технології польоту-часу дозволяє отримати глибину зображення (рис. 2.5). Камера поєднує в собі 940-нм інфрачервоний вертикальний лазер з поверхневим випромінюванням і датчик зображення з роздільною здатністю 240180 пікселів, який технічно здатний знімати 120 кадрів в секунду, але на Raspberry Pi обмежений до 30 кадрів в секунду. Датчик, який має 70-градусне поле зору, може збирати дані про глибину до 2 м футів у "ближньому режимі" і до 5 м у "дальньому режимі" [20].



Рисунок 2.5 – Зовнішній вигляд ArduCam ToF Camera

Камера розміром всього 38×38 мм підключається до порту MIPI Camera Serial Interface (CSI) Raspberry Pi, а також до універсального заголовка вводу/виводу (GPIO) для живлення лазера і пропонує зразки коду на мовах C, C++ і Python для збору необроблених даних і обробки глибини. Драйвери Arducam також розкривають датчик, що робить його сумісним з фреймворками комп'ютерного зору та машинного навчання, такими як OpenCV і TensorFlow.

Що стосується точності, то сенсор має точність до 2 см в дальньому режимі, тому ідеально підходить для додатків, які потребують базового розуміння глибини різкості[21].

За допомогою камери Arducam ToF також можна створити хмару точок. Це набір точок даних у просторі, які утворюють тривимірну форму або об'єкт. Для захоплення глибини більшості хмар точок потрібен інструмент 3D-сканування або камера з подвійним об'єктивом. Поєднання камери Arducam ToF з Raspberry Pi є чудовим методом побудови недорогого 3D-сканера, який фіксує фактичні дані про глибину, а не намагається оцінити її на основі взаємного розташування об'єктів на фотографіях. Характеристику ArduCam ToF Camera наведено у табл. 2.5

Кут зору	70°
Максимальна частота кадрів	120 к/с
Напруга живлення	3.5 - 5 В
Протокол	I2C
АЦП	16-бітний
Розміри	38 x 38 мм
Довжина хвилі лазера	940-нм
Дальність дії	5 м

Таблиця 2.5 – Характеристики ArduCam ToF Camera

Через велику кількість природного інфрачервоного світла від сонця, інфрачервоні камери зазвичай має труднощі при роботі на вулиці. Однак ArduCam оснастили цю камеру лазером, що працює на довжині хвилі 940 нанометрів, який здатний придушити перешкоди від навколишнього світла на вулиці досить добре, щоб забезпечити досить хороші результати, особливо при спостереженні за великими об'єктами з простою геометрією, такими як великі коробки або людські фігури[22].

Камера працює як стандартний пристрій V4L2 або Video4Linux2, тому ви можете використовувати її з низкою бібліотек технічного зору, таких як OpenCV і Tensorflow, без необхідності робити багато додаткового кодування. камеру Arducam Time of Flight можливо використовувати з іншими дистрибутивами, оскільки вони мають набори для розробки на C, C++ і Python[23].

VL53L1X Waveshare - це модуль вимірювання дальності за часом польоту (ToF) на базі VL53L1X від ST, з точною дальністю до 4 м і швидкою частотою вимірювання до 50 Гц, він управляється через інтерфейс I2C і має досить низьке енергоспоживання (рис. 2.6).

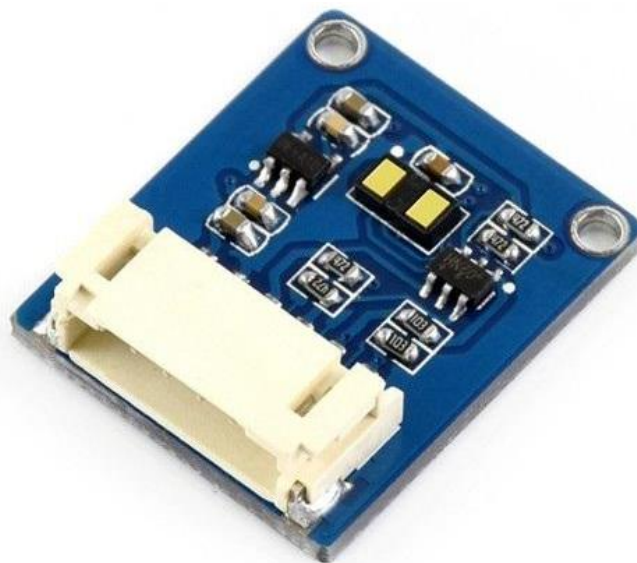


Рисунок 2.6 – Зовнішній вигляд VL53L1X Waveshare

На відміну від традиційних датчиків дальності, VL53L1X містить фізичні інфрачервоні фільтри та оптику, використовує технологію ST поточного покоління ToF для вимірювання абсолютної відстані незалежно від кольору та відбивної здатності цілі, а також забезпечує кращі показники захисту від перешкод. У таблиці 2.6 наведено технічні характеристики модулю VL53L1X Waveshare[24].

Кут зору	27°
Максимальна частота кадрів	120 к/с
Напруга живлення	3.3 - 5 В
Протокол	I2C
АЦП	16-бітний
Розміри	20 x 24 мм
Довжина хвилі лазера	940 нм

Таблиця 2.6 – Характеристики VL53L1X Waveshare

Комунікаційний інтерфейс I2C, модуль керування ввімкненням/вимкненням за допомогою контактів вводу/виводу. Вбудований перетворювач напруги з підтримкою робочої напруги 3,3В/5В. У таблиці 2.6 наведено технічні характеристики модулю VL53L1X Waveshare[25].

1.2 Опис програмного середовища Fritzing

Fritzing - це програмне забезпечення з відкритим вихідним кодом, яке робить електроніку доступною для будь-кого (рис. 2.7). Це електронний інструмент автоматизації проектування для неінженерів. Fritzing дозволяє користувачам проектувати друковані плати і надає доступ до комерційного сервісу під назвою "FritzingFab" для замовлення друкованих плат, створених за допомогою дизайну Fritzing.

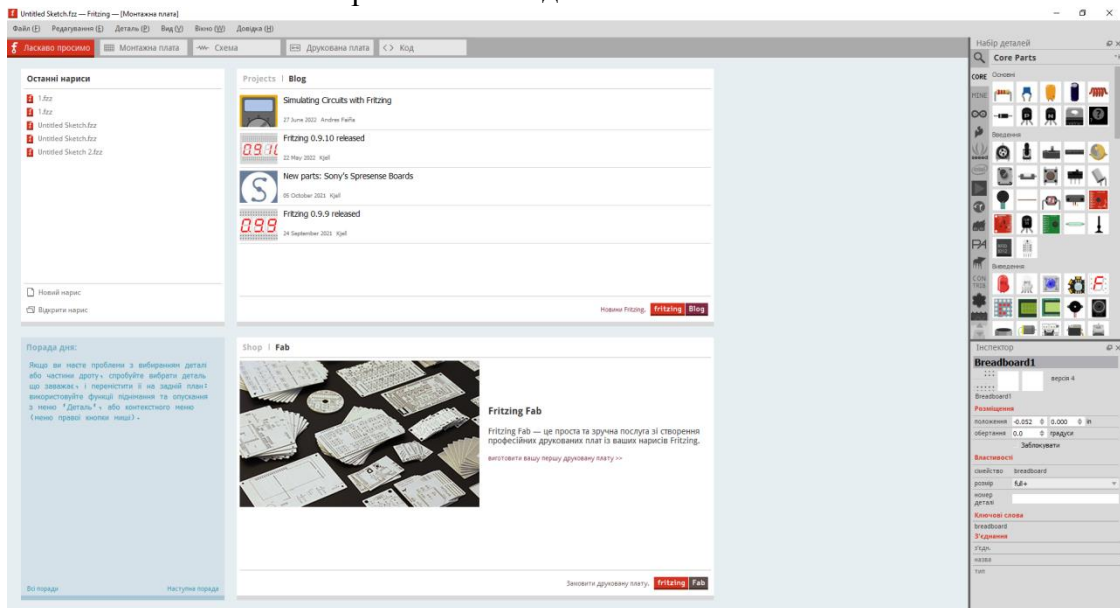


Рисунок 2.7 – Інтерфейс програмного середовища Fritzing

Опис основних елементів інтерфейсу програмного забезпечення Fritzing:

1. Вкладка «Ласкаво просимо» – перша вкладка, що вітає користувача. На данній вкладці користувач може знайти «пораду дня», що надає корисну пораду що до використання програмного забезпечення. Актуалбні новини щодо Fritzing. Також у даному вікні знаходиться список останніх створених проєктів.

2. Вкладка «Монтажна плата» (рис. 2.8) – наступна вкладка з макетною платою в якій створюється макетна схема.

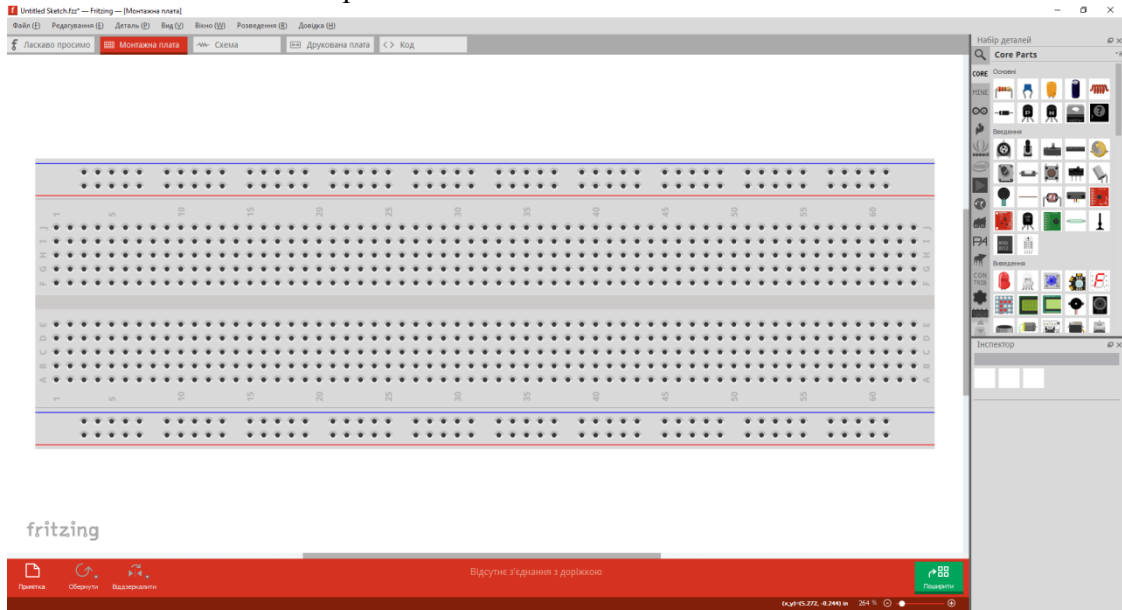


Рисунок 2.8 – Вкладка «Монтажна плата»

У правій частині розміщено вікно у якому користувач можете знайти та розмітити необхідні компоненти у вкладці «Монтажна плата» (рис. 2.9).

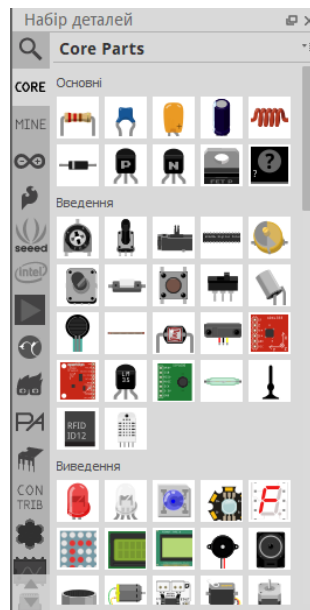


Рисунок 2.9 – Вкладка «Монтажна плата»

3. Вкладка «Схема» (рис. 2.10) – третя вкладка в якій автоматично створюється принципова схема проекту. Коли користувач створює схему макетної плати, Fritzing автоматично додає деталі до схеми і з'єднує їх на основі їх розміщення на макетній платі.

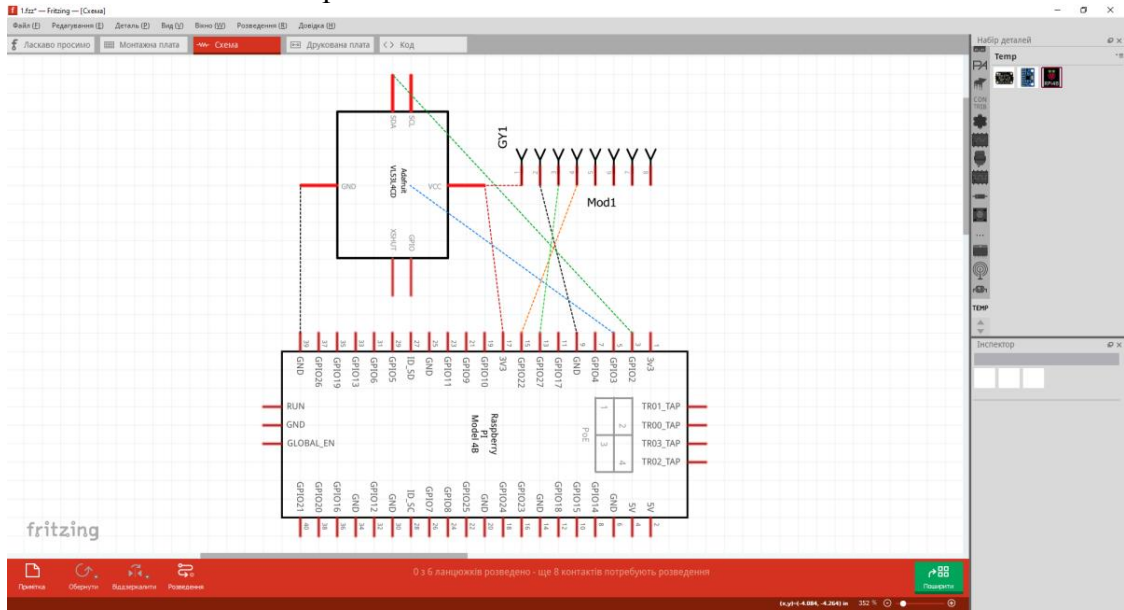


Рисунок 2.10 – Вкладка «Монтажна плата»

4. Вкладка «Друкована плата» (рис. 2.11) – вкладка, де створюється друкована плата. Компонування друкованої плати є завершальним етапом у процесі створення схеми Фрітцинга. Частина плати автоматично розміщуються на друкованій платі, так само, як і на етапі проектування схеми. Однак, користувачу необхідно правильно розташувати всі компоненти. Оскільки не кожен проект потребує друкованої плати, етап створення друкованої плати є необов'язковим. Однією з основних переваг використання Fritzing є те, що програмне забезпечення працює в тандемі з послугою Fritzing PCB, яка швидко перетворює дизайн Fritzing в недорогу друковану плату.

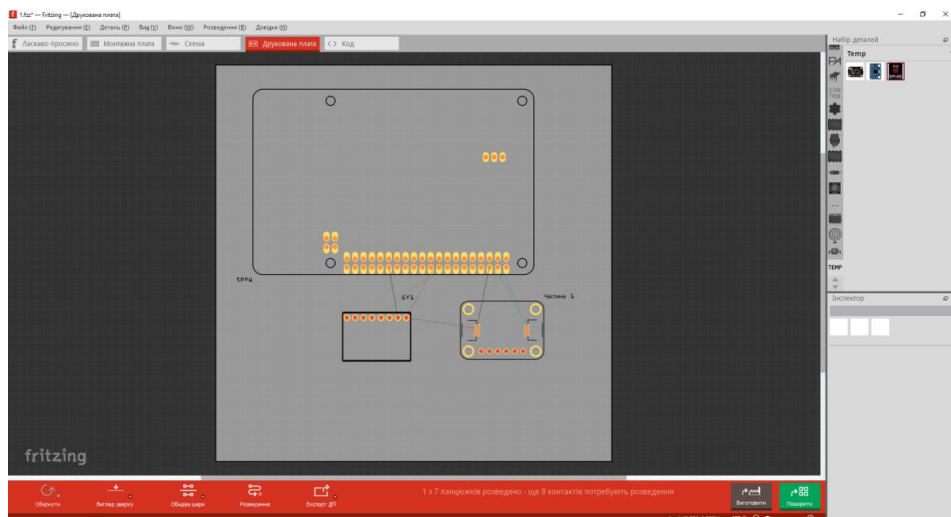


Рисунок 2.11 – Вкладка «Друкована плата»

5. Вкладка «Код» (рис. 2.12) – вкладка в якій користувач пише програмний код. Знизу розміщені кнопки, що відповідають за створення, відкриття та зберігання файлу з кодом, кнопки серійного монітору та завантаження. Також наявні випадючі вікна за допомогою яких можна обрати платформу, плату, комунікаційний порт.

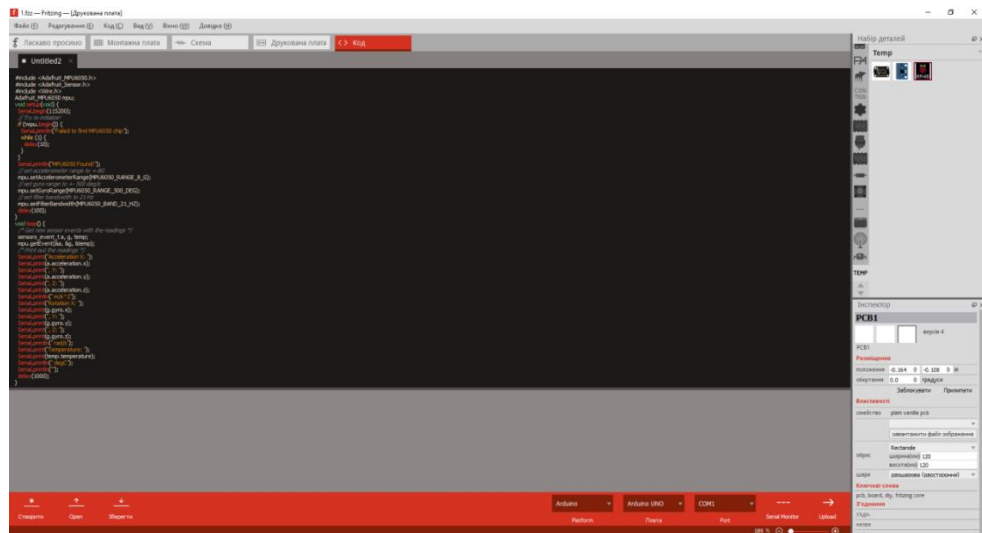


Рисунок 2.11 – Вкладка «Код»

Висновки до розділу 2

В цьому розділі було проведено огляд компонентів апаратно-програмного комплексу, а саме було вибрано мікроконтролер Raspberry Pi 4 Model B, гіроскопічний датчик MPU-6050, та ToF камеру ArduCam. Було розглянуто основні програмні забезпечення.

3 РОЗРОБКА АПАРАТНО-ПРОГРАМНОГО КОМПЛЕКСУ

2.1 Алгоритм роботи системи real time симуляція руху об'єкту

Система прикріплюється бажаного об'єкту, що здатний переміщуватися, таких як: дрони (повітряні, наземні, наводні, підводні), транспорт, маніпулятори, м'ягка робототехніка. ToF камера проводить сканування місцевість, чи об'єкти у режимі реального часу. За допомогою гіроскопічного, акселерометричного датчика MPU-6050 відбувається калібрування зсуву, повороту та переміщення пристрою. Отримані дані відправляються на обробку на сервер. Вже оброблені данні повертаються на пристрій користувача, звідки вже користувач може керувати пристроєм та сканувати місцевість. При необхідності користувач може здійснити до режиму альтернативного режиму користування «вільного польоту», де може незалежно від системи навігації власноруч переміщуватися та вивчати відскановане середовище у віртуальному просторі, або ж у доповненій реальності.

2.2 Створення блок-схеми алгоритму визначення переміщення

На рис. 3.1 зображено блок-схему алгоритму визначення переміщення. Спочатку зчитуються данні про початкове положення пристрою у просторі. Далі здійснюється затримка у 100 мс. Після затримки пристрій считує нові данні про положення системи у просторі. Маючи нові данні відбувається розрахунок різниці між початковими та кінцевими даними. В кінці відправляються данні про переміщення і алгоритм повторюється з початку.



Рисунок 3.1 – Блок-схеми алгоритму визначення переміщення

2.3 Проектування системи real time симуляції руху об'єкту на базі гіроскопічного датчика MPU-6050

Система real time симуляції руху об'єкту складається з шестиосьового датчика MPU-6050, камери часу польоту ArduCam та Raspberry Pi 4 Model B. Живлення до камери часу польоту ArduCam подається через пін 1 на Raspberry Pi 4 Model B. Земля під'єднується до пину 9 на платі Raspberry Pi 4 Model B. Зображення з ArduCam передається у спеціальний 2-лінійний MIPI CSI порт. Для передачі даних підключаємо (SDA) пін датчика MPU-6050 до Raspberry Pi 4 Model B на пін 15. Тактові імпульси (SCL) передаються на пін 13. Живлення та земля передаються через Raspberry Pi 4 Model B по пінам 17 та 20 відповідно. На рис. 3.2 зображено макетну схему, а на рис. 3.3 принципову схему акустичного комплексу.

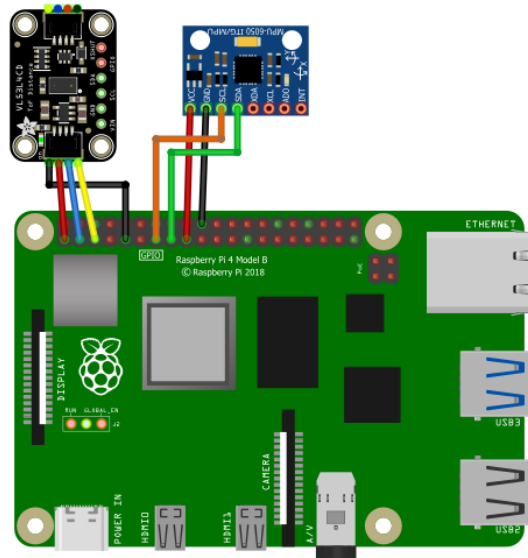


Рисунок 3.2 – Макетна схема системи real time симуляції руху об'єкту

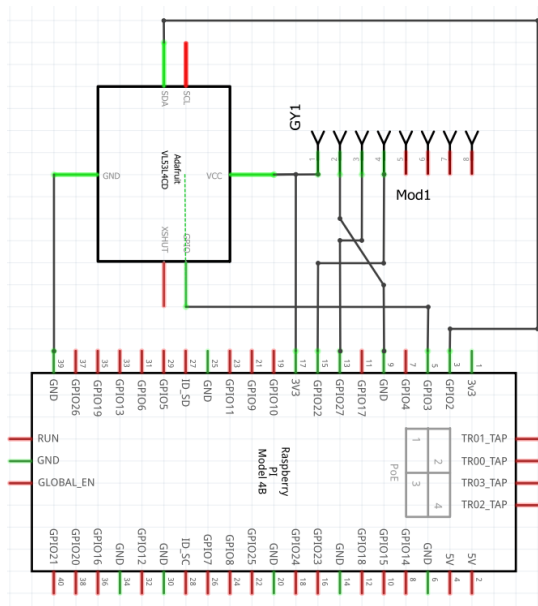


Рисунок 3.3 – Принципова схема системи real time симуляції руху об'єкту

2.4 Проектування концептуальної схеми апаратно-програмного комплексу

Концептуальна схема, також відома як концептуальна модель, - це високорівневе представлення системи або предметної області, яке фіксує ключові поняття, сутності та зв'язки між ними. Це спосіб організації та

Кафедра комп'ютерної інженерії
Апаратно-програмний комплекс для real-time симуляції руху об'єкту на базі
гіроскопічного датчика MPU-6050
структурування інформації таким чином, щоб допомогти людям розуміти і
міркувати про систему або предметну область.

Концептуальна схема зазвичай використовується на ранніх стадіях проектування або моделювання системи, до початку детальної реалізації або будівництва. Вона забезпечує спільне розуміння системи або предметної області для всіх зацікавлених сторін, включаючи проектувальників, розробників, користувачів та інших зацікавлених осіб.

Концептуальні схеми можуть бути представлені за допомогою різних методів моделювання, таких як діаграми UML, діаграми "сутність-зв'язок" (ER) та інші подібні інструменти. Вони можуть бути використані для визначення вимог, виявлення потенційних проблем або викликів, а також для вивчення різних альтернативних варіантів проектування. Після узгодження концептуальної схеми вона може слугувати основою для більш детального проектування та реалізації системи.

Всю схему можна умовно поділити на три частини. Перша частина описує процес сканування. Апаратно-програмного комплекс поміщається у певне середовище да сканує певні сутності будь то приміщення, чи певний об'єкт (рис. 3.5).

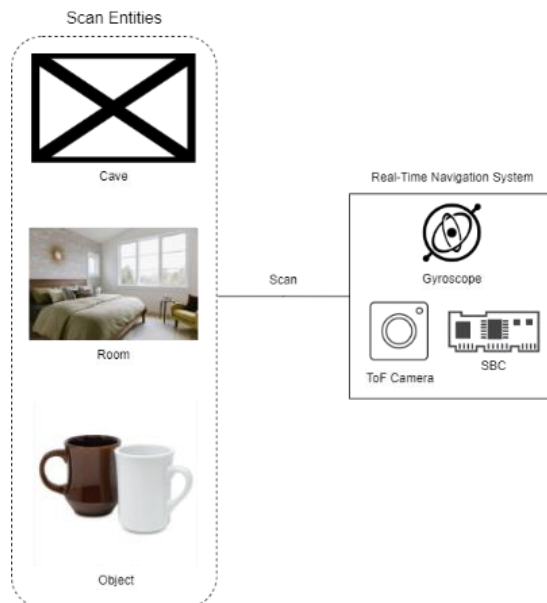


Рисунок 3.4 – Сканування сутностей

Наступним кроком апаратно-програмний комплекс надсилає дані про мапу глибин та переміщення на сервер. На рис. 3.5 зображена серверна частина, в якій сервер оброблює отриманні дані та створює 3D сцену. Сервер має в собі два сховища. В перше сховище він відправляє файл з 3D сценою, та данні про оновлення 3D сцени. У другому сховище надсилаються та зберігаються мета данні про 3D оточення.

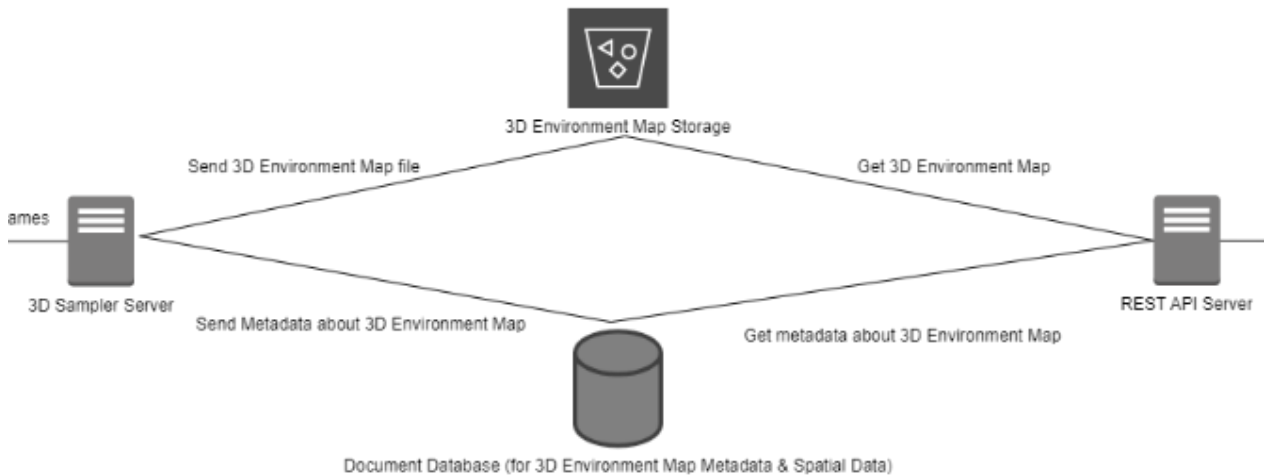


Рисунок 3.5 – Серверна частина

Третя частина зображена на рисунку 3.6 передає данні через REST API. REST API (Representational State Transfer Application Programming Interface) - це інтерфейс прикладного програмування, який відповідає обмеженням архітектури REST. Він забезпечує безпечний зв'язок між двома комп'ютерними системами через Інтернет. REST API не залежать від технології і можуть бути реалізовані різними способами. Вони складаються з набору взаємопов'язаних ресурсів, запити до яких обробляються через HTTP. Клієнт і сервер можуть взаємодіяти лише одним способом: клієнт надсилає запит на сервер, а сервер відповідає. REST API адаптуються і масштабуються, що робить їх ідеальними для веб-додатків, які повинні обробляти широкий спектр запитів і надавати дані в різних форматах. Вони також використовують сучасні веб-технології, такі як HTTP, що робить їх більш зручними для користувача, ніж спеціальні протоколи, такі як SOAP (Simple Object Access Protocol). Дані передаються на такі пристрої як:

Кафедра комп'ютерної інженерії
Апаратно-програмний комплекс для real-time симуляції руху об'єкту на базі
гіроскопічного датчика MPU-6050
мобільні телефони, ноутбуки, персональні комп'ютери та пристрої XR
(VR/AR).

PA
GE
\
*
M
ER
GE
FO
R
M

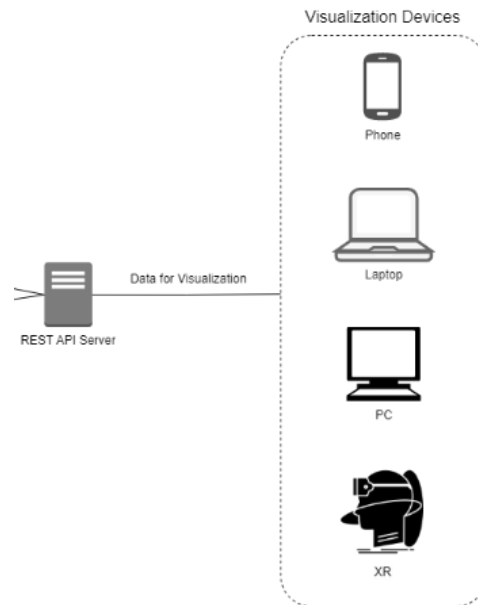


Рисунок 3.5 – Передача 3D середовища на пристрій візуалізації

2.5 Розробка структурної діаграми схеми апаратно-програмного комплексу

Структурна діаграма - це тип інструменту концептуального моделювання, який використовується для документування різних структур, що складають систему, наприклад, базу даних або додаток. Вона відображає ієрархію або структуру різних компонентів або модулів системи, а також те, як вони пов'язані та взаємодіють один з одним. Структурні діаграми, також відомі як структурні моделі UML, відображають елементи моделі незалежно від часу, тоді як діаграми технічної архітектури вимагають створення абстрактного опису сутностей у системі та їхніх зв'язків між собою.

На рис. 3.6 зображено технічну частину діаграми, дана частина складається з ToF камери, 3-осьового гіроскопічного та 3-осьового акселерометричного датчика, мікроконтролера котрий запрограмований мовою Python.

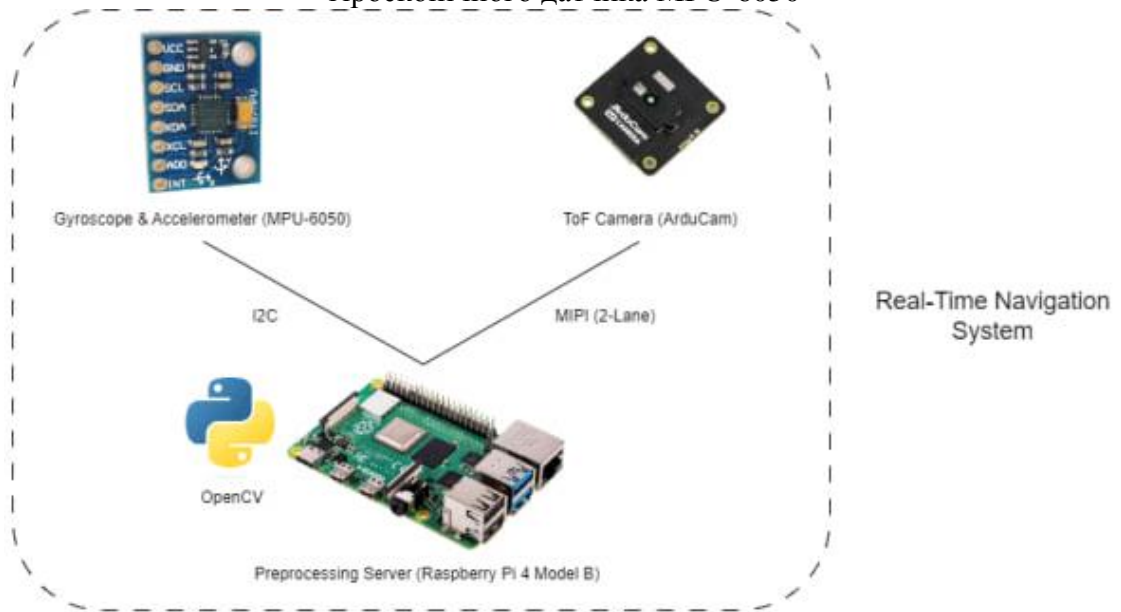


Рисунок 3.6 – Технічна частина діаграми

Далі зображена серверна частина (рис. 3.7) на яку передаються дані через інтернет за допомогою ZeroMQ. ZeroMQ, також відома як ØMQ, - це бібліотека обміну повідомленнями з відкритим вихідним кодом, яка дозволяє програмним додаткам спілкуватися один з одним у гнучкий та ефективний спосіб. ZeroMQ розроблена як легка, швидка і масштабована, що робить її популярним вибором для розподілених систем і додатків реального часу.

ZeroMQ базується на концепції передачі повідомлень, де програми спілкуються, надсилаючи та отримуючи повідомлення. Він надає простий, але потужний API, який дозволяє програмам надсилати та отримувати повідомлення, використовуючи різні шаблони обміну повідомленнями, такі як публікація/підписка, запит/відповідь та push/pull[26].

ZeroMQ підтримує широкий спектр мов програмування, включаючи C, C++, Java, Python, Ruby та багато інших, і може використовуватися на різних операційних системах, включаючи Linux, Windows і macOS.

ZeroMQ став популярним вибором для побудови розподілених систем і мікросервісів, і використовується багатьма організаціями для різноманітних додатків, включаючи фінансову торгівлю, ігри, соціальні мережі тощо.

Кафедра комп'ютерної інженерії
Апаратно-програмний комплекс для real-time симуляції руху об'єкту на базі
гіроскопічного датчика MPU-6050
зберігає в собі 3D середовища. У другій базі даних на Amazon DynamoDB
зберігаються метаданні про ці 3D середовища.

Amazon Simple Storage Service (Amazon S3) - це сервіс зберігання об'єктів, який пропонує найкращі в галузі масштабованість, доступність даних, безпеку та продуктивність. Він зберігає дані у вигляді об'єктів у кошиках і пропонує різноманітні типи сховищ, які підходять для різних сценаріїв використання.

Amazon S3 також включає опції управління сховищем, які допомагають контролювати витрати, відповідати нормативним вимогам, зменшувати затримки та зберігати багато окремих копій даних для цілей дотримання нормативних вимог. Клієнти можуть оцінити свої витрати за допомогою калькулятора цін AWS і оплатити функції управління сховищем та аналітики (Amazon S3 Inventory, S3 Storage Class Analysis, S3 Storage Lens і S3 Object Tagging).

Функції Amazon S3 включають можливість додавати теги метаданих до об'єктів, переміщувати та зберігати дані в різних класах сховищ S3, налаштовувати та впроваджувати політики доступу до даних, розміщувати веб-сайти та зберігати великі обсяги даних за низькою ціною.

Amazon DynamoDB - це повністю керована власна служба баз даних NoSQL від Amazon Web Services (AWS). Він підтримує структури даних типу ключ-значення та документ і пропонує швидку та передбачувану продуктивність, а також плавне масштабування. Вона ідеально підходить для безсерверних додатків, які використовують Amazon Lambda, наборів даних з простими відомими шаблонами доступу та додатків, які вимагають високої пропускної здатності з низькою затримкою.

Amazon DynamoDB може зберігати та отримувати будь-яку кількість даних, обслуговувати будь-який рівень трафіку запитів і масштабувати свою пропускну здатність вгору або вниз. Вона також має автоматичне масштабування, ефемерні обчислення без стану у відповідь на тригери подій,

реплікацію між різними центрами обробки даних для забезпечення надійності та доступності, динамічну побудову таблиць та інші можливості.

Amazon DynamoDB - це власна служба баз даних NoSQL, яка повністю керована і підтримує типи даних "ключ-значення" та "документ". Amazon.com надає її як частину інфраструктури Amazon Web Services (AWS). DynamoDB відрізняється від інших сервісів Amazon тим, що дозволяє розробникам платити за послугу на основі пропускної здатності, а не обсягу пам'яті.

Елементи, як і рядки в реляційних базах даних, представляють окремі об'єкти в таблиці. Атрибути - це фундаментальні одиниці інформації, такі як пари ключ-значення, які складають елемент. Надаючи різні методи доступу до одних і тих самих даних, індекси дозволяють користувачам ефективніше робити запити до даних.

Потоки DynamoDB забезпечують впорядковану в часі послідовність змін, внесених до даних таблиці, що дозволяє розробникам отримувати та оновлювати дані на рівні елементів до і після внесення змін. Як наслідок, це корисно для таких операцій, як аналітика та аудит в режимі реального часу. Крім того, DynamoDB регулярно створює резервні копії даних користувачів з міркувань безпеки, дозволяючи власникам зберігати свої дані в хмарі для веб-сайтів або мобільних додатків[28].

DynamoDB використовується для різних випадків використання, таких як підвищення цінності для клієнтів за допомогою хмарного доступу, зберігання та обчислень; зберігання маркетингових даних різних типів; та надання аналітики в реальному часі на основі операційних даних. Вона також може бути використана для таких додатків, як управління даними профілю користувача або інформацією про веб-сесії для веб-сайтів або мобільних додатків.

Далі ідуть сервара на REST API, що використовують m7g.large, які також можуть бути горизонтально розширені. Інстанси M7g на базі

процесорів AWS Graviton3 поточного покоління пропонують найкраще співвідношення ціни та продуктивності в Amazon EC2 для робочих навантажень загального призначення. Екземпляри M7g підходять для програм з відкритим вихідним кодом, таких як сервери додатків, мікросервіси, ігрові сервери, сховища даних середнього розміру та парки кешування. Вони перевершують екземпляри M6g шостого покоління на базі Amazon Graviton2 на 25%. Примірники M7g оснащені пам'яттю Double Data Rate 5 (DDR5), яка забезпечує на 50% більшу пропускну здатність, ніж пам'ять DDR4, і дає змогу пришвидшити доступ до даних у пам'яті.

В кінці серверної частини йде Балансувальник навантаження додатків. Балансувальник навантаження додатків - це функція еластичного балансування навантаження, яка дозволяє розробнику налаштовувати і направляти вхідний трафік кінцевих користувачів до додатків, що базуються в публічній хмарі AWS. Він працює на сьомому рівні моделі взаємодії відкритих систем (OSI), відомому як прикладний рівень.

Щоб підвищити доступність, балансувальник навантаження додатків розподіляє вхідний трафік додатків між кількома цілями, наприклад, екземплярами EC2, у декількох зонах доступності. Він перевіряє пакети і створює точки доступу до заголовків HTTP і HTTPS, розпізнаючи тип навантаження і розподіляючи його на ціль з більшою ефективністю в залежності від таких параметрів, як затримка, стан та інші.

Переваги використання балансувальника навантаження додатків включають покращену продуктивність балансувальника навантаження, підвищення доступності та масштабованості додатків, автоматичне перемикання між серверами в разі простою або низької продуктивності, а також детальні звіти, що містять такі метрики, як коди HTTPS і затримки.

2.6 Розробка діаграми послідовності апаратно-програмного комплексу

Діаграми послідовності - це тип діаграм UML (Unified Modeling Language), які візуально представляють взаємодію між об'єктами в системі. Вони показують порядок взаємодії та повідомлення, які передаються між об'єктами.

На діаграмі послідовності об'єкти, що беруть участь у взаємодії, зображуються вертикальними лініями, які називаються "лініями життя". Взаємодія між об'єктами показана горизонтальними стрілками або повідомленнями, причому стрілка вказує від відправника до одержувача повідомлення. Повідомлення можуть також містити інформацію про тип повідомлення, будь-які параметри, що передаються, та значення, що повертається, якщо таке є.

Діаграми послідовності часто використовуються в програмній інженерії та системному дизайні для моделювання взаємодії між різними компонентами системи, а також для виявлення потенційних проблем або сфер, які потребують вдосконалення в дизайні системи. Їх можна використовувати на різних етапах розробки, від раннього проектування до впровадження, і вони можуть бути корисними для пояснення поведінки системи як технічним, так і нетехнічним зацікавленим сторонам.

На рис 3.8 зображено діаграму послідовності на якій розміщено такі елементи – гіроскоп і кселерометр, сервер попередньої обробки, ToF камера, балансувальник, сховище 3D-карт навколишнього середовища, сервер 3D-семплеру, сервер REST API, база даних для метаданих 3D мап довкілля, балансувальник, пристрій візуалізації.

Кафедра комп'ютерної інженерії
 Апаратно-програмний комплекс для real-time симуляції руху об'єкту на базі
 гіроскопічного датчика MPU-6050

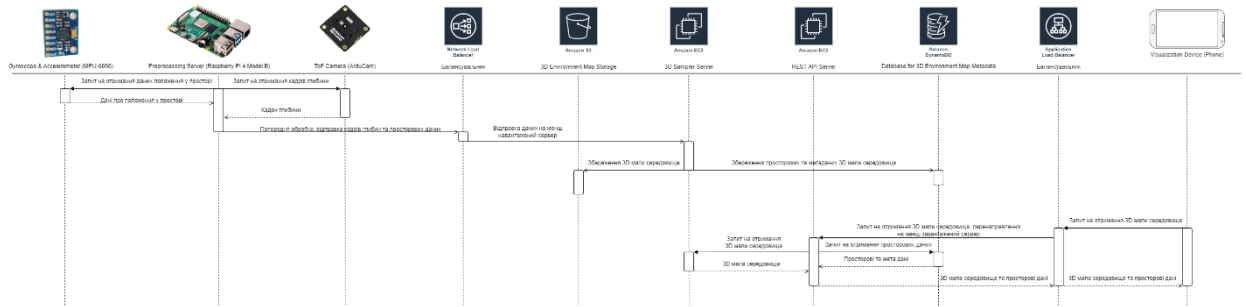


Рисунок 3.8 – Діаграма послідовності апаратно-програмного комплексу

Система працює таким чином, сервер попередньої обробки робить запит до датчика гіроскопа і акселерометра, той у відповідь відправляє данні про положення у просторі та його зміну. Далі Raspberry Pi 4 створює запит до ToF камери, яка у свою чергу передає данні про мапу глибини. Наступним кроком мікроконтролер робить попередню обробку мапи глибини та просторові данні та надсилає її до балансувальника. Балансувальник обирає та передає на найменш навантажений сервер отриманні данні. Сервер 3D-семплеру зберігає 3D мапу, паралельно відбувається збереження просторових та метаданих 3D середовища. Коли пристрій візуалізації віправляє запит на отримання 3D середовища, на що балансувальник створює запит на отримання 3D мапи середовища, перенаправлення на менш завантажений сервер. У відповідь REST API сервер створює 2 запити. Перший запит адресується до серверу 3D-семплеру на отримання 3D мапи середовища. Другий запит на отримання просторових даних до бази даних для метаданих 3D мап довкілля. У відповідь перша база даних відправляє 3D мапу середовища, в друга надає просторові метадані. В результаті REST API сервер відправляє до балансувальника 3D мапу середовища та просторові дані, а далі вони відправляються до пристрою візуалізації.

2.7 3D сканування за допомогою використання мапи глибин

Більшість камер ToF працюють у двох довжинах світла, або у 850 нм, або ж у 940 нм. Камери, що передбачають роботу у навколишньому

Кафедра комп'ютерної інженерії
Апаратно-програмний комплекс для real-time симуляції руху об'єкту на базі
гіроскопічного датчика MPU-6050
середовищи працюють на довжині 940 нм, через природне зниження
сонячного світла в цьому діапазоні (рис. 3. 9).

PA
GE
*
M
ER
GE
FO
R
M

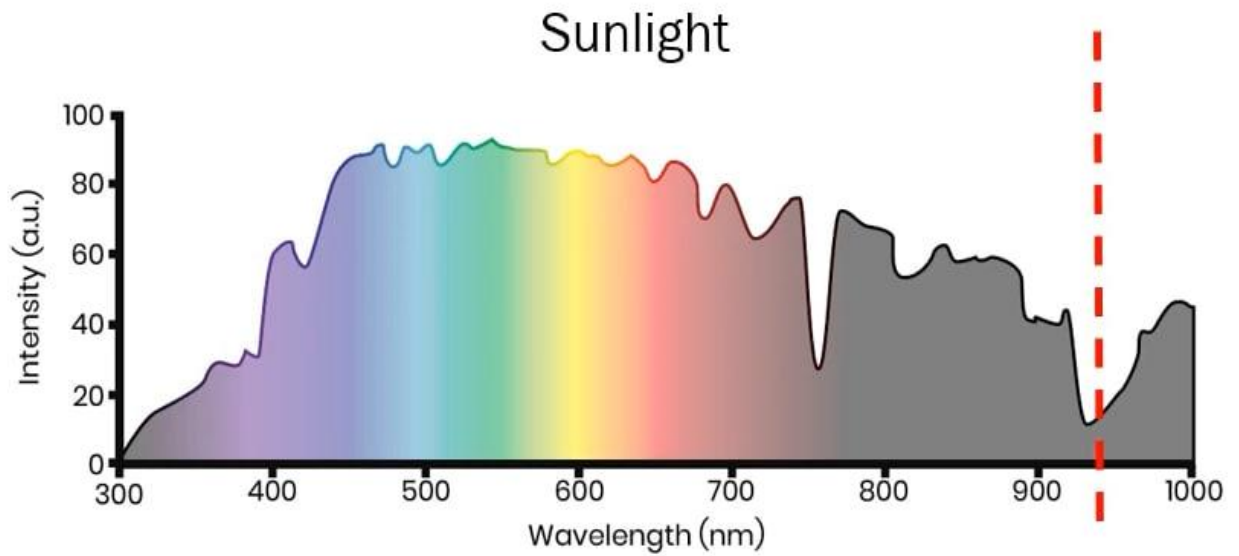


Рисунок 3.9 – Крива спектрального відгуку

Більшість зовнішніх датчиків мають смуговий фільтр для передбачення
перешкоджання роботи датчика іншою частиною сонячного світла (рис.
3.10).

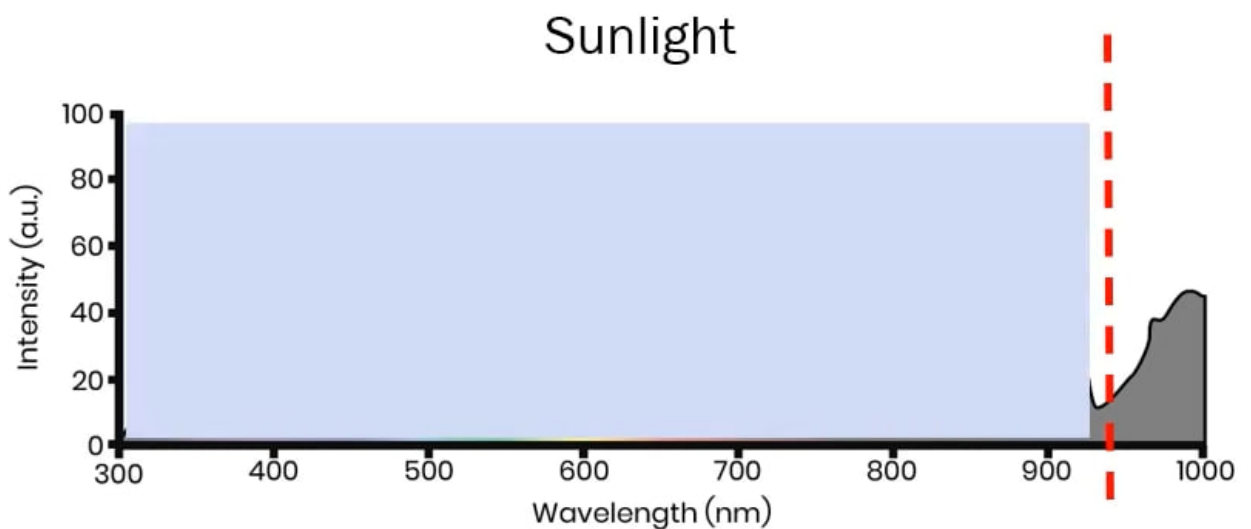


Рисунок 3.9 – Смуговий фільтр для ToF камер

У разі відсутності подібного фільтру, яскравість сонячного світла
засвітить зображення, що спричинить до неможливості вилучення корисної
інформації із зображення.

Підчас роботи ToF камера видає мапу глибини у виді кольорового зображення, чи чорно-білого, де червоний/білий це найближча дистанція, а синій/чорний найвіддаленіша (рис 3.10).

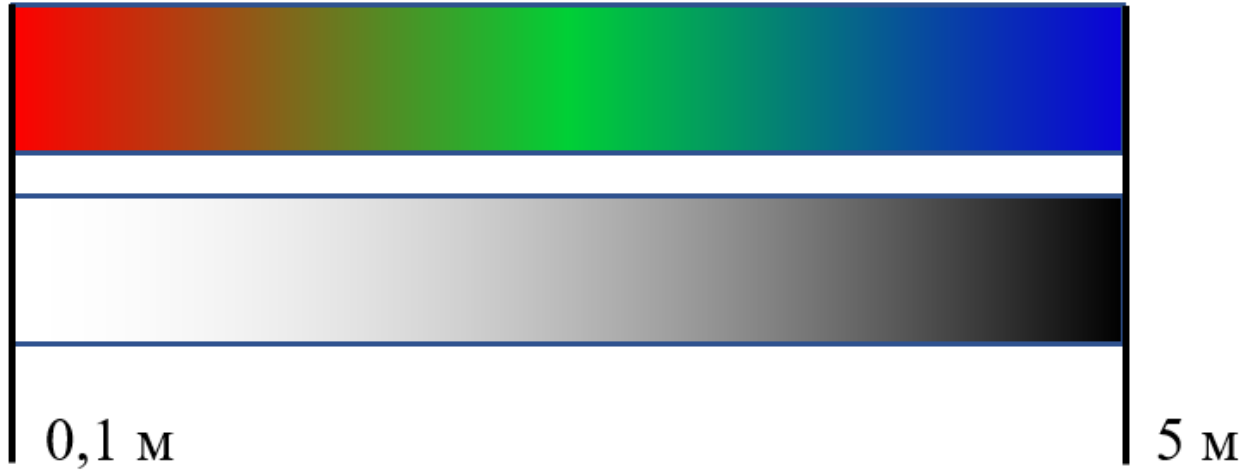
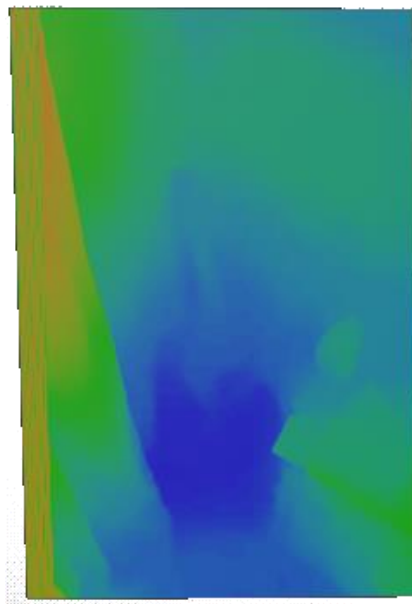


Рисунок 3.9 – Кольорове представлення дистанції у маві глибини

На рис. 3.10 отриманий знімок з ToF камери має детальне зображення кожен піксель якого можна представити у числовому вигляді. В результаті маємо матрицю чисел, де кожне число це дистанція до точки. Далі перетворюємо ці точки у вокселі.



Воксель - це тривимірний піксель, який представляє значення або атрибут у певній точці тривимірного простору. Слово "воксель" - це поєднання слів "об'єм" і "піксель". Подібно до пікселя, який представляє значення в точці двовимірного простору, воксель представляє значення в точці тривимірного простору.

Вокселі часто використовуються для представлення об'ємних даних, таких як медичні зображення з КТ, МРТ або ультразвукового сканування, 3D-моделі з 3D-сканерів або програмного забезпечення САПР, або дані з симуляцій чи наукових експериментів. У воксельному представленні 3D-простір поділяється на регулярну сітку маленьких кубічних комірок, кожна з яких відповідає вокселю. Кожен воксель зберігає значення або атрибут, наприклад, щільність, інтенсивність, колір або матеріал, у відповідному місці в просторі.

Вокселі мають багато переваг над іншими формами представлення, такими як хмари точок або сітки, у певних додатках. Наприклад, вокселі можуть забезпечити більш точне і повне представлення об'ємних даних і можуть використовуватися для різних операцій, таких як фільтрація, сегментація або реєстрація, які вимагають об'ємної інформації. Однак вокселі також можуть бути дорогими в обчислювальному плані та вимагати багато пам'яті, особливо для даних з високою роздільною здатністю або великих обсягів.

Попередньо необхідно обробити мапу глибини: мапа глибини, отримана з камери ToF, може містити шум, прогалини або відсутні дані. Щоб підготувати карту глибини для алгоритмів об'ємної реконструкції, вам може знадобитися відфільтрувати дані, заповнити пропуски або інтерполювати відсутні значення.

Перетворюємо мапи глибини у воксельну сітку. Для подібного перетворення можна використати два алгоритми: *Marching Cubes*, або

Plenoxels які потребують воксельної сітки як вхідних даних. Є можливість перетворити карту глибини у воксельну сітку, дискретизувавши значення глибини у звичайну 3D-сітку вокселів. Both Marching Cubes and Plenoxels are volumetric reconstruction algorithms that can generate a 3D voxel map from a depth map. I Marching Cubes, i Plenoxels - це алгоритми об'ємної реконструкції, які можуть генерувати 3D воксельну карту з карти глибини. Нижче наведено загальні кроки для використання маршируючих кубів та плінокселів для створення воксельної 3D-карти

Застосування алгоритму реконструкції. Користувач у застосунку запускає алгоритм Marching Cubes або Plenoxels на воксельній сітці, щоб згенерувати об'ємне представлення сцени (рис. 3.11). Алгоритм Marching Cubes витягує поверхневу сітку, яка апроксимує форму об'єктів в об'ємі, в той час як алгоритм Plenoxels генерує імовірнісну сітку зайнятості, яка присвоює кожному вокселю значення ймовірності, що вказує на те, чи зайнятий воксель чи ні.

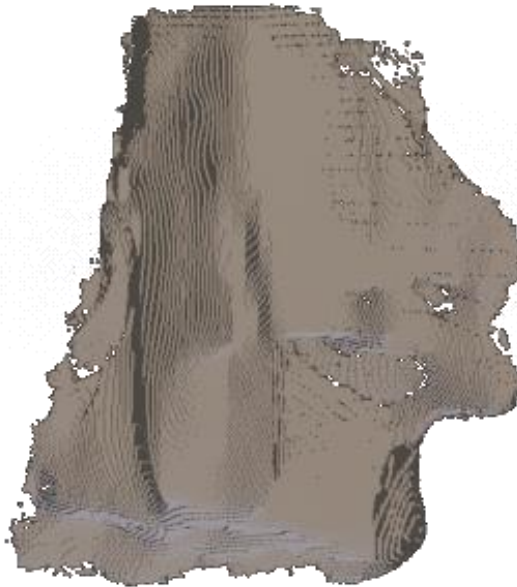


Рисунок 3.11 – Воксельне представлення 3D сцени

Постобробка воксельної сітки. Після створення воксельної сітки необхідно просвети постобробку, щоб видалити артефакти, згладити межі та створити полігональну сітку (рис. 3. 12).



Рисунок 3.12 – Полігональне представлення 3D сцени

Вибір між алгоритмом Marching Cubes та Plenoxels залежить від конкретних вимог застосунку. Marching Cubes більше підходить для застосунків, які потребують полігональної сітки об'єкта або середовища з високою роздільною здатністю, в той час як Plenoxels більше підходить для додатків, які потребують імовірнісної сітки заповнення середовища, наприклад, роботизованої навігації або доповненої реальності.

В цілому, і Marching Cubes, і Plenoxels є потужними алгоритмами об'ємної реконструкції, які можуть генерувати 3D воксельну карту з карти глибини. Ви можете вибрати алгоритм, який найкраще підходить для вашої програми, і експериментувати з різними параметрами, щоб досягти бажаного рівня деталізації і точності.

Marching Cubes - це популярний алгоритм для вилучення сітки поверхні з об'ємного зображення об'єкта або середовища. Алгоритм був вперше представлений у 1987 році Вільямом Е. Лоренсеном і Харві Е. Клайном і з тих пір широко використовується в різних додатках, таких як медична візуалізація, комп'ютерна графіка і 3D-друк.

Нижче наведено детальний опис алгоритму Marching Cubes:

1. Вокселізація: Першим кроком алгоритму Marching Cubes є вокселізація вхідних даних, які можуть бути 3D-масивом скалярних значень або імовірнісною сіткою заповнення. Процес вокселізації дискретизує вхідні дані у звичайну 3D-сітку вокселів з рівномірним розміром.

2. Підрозділ кубів: Алгоритм обробляє кожен воксель у сітці та досліджує скалярні значення навколишніх вокселів, щоб визначити, чи знаходиться воксель всередині або зовні поверхні. Для цього алгоритм розбиває кожен воксель на вісім менших кубів, кожен з яких має власний набір скалярних значень.

3. Скалярна класифікація: Для кожного з восьми кубів алгоритм оцінює скалярні значення восьми вершин і визначає, чи лежать вони всередині або зовні поверхні. Результатом цієї класифікації є двійковий код з вісьмома бітами, що представляє конфігурацію куба. Всього існує 256 таких комбінацій, які можна утворити, дивлячись на значення наших вершин, оскільки кубики мають 8 кутів з кожними 2 можливими станами ($2^8=256$). Ці 256 конфігурацій можна звести до 15, оскільки більшість випадків є симетріями (рис3.13)[29].

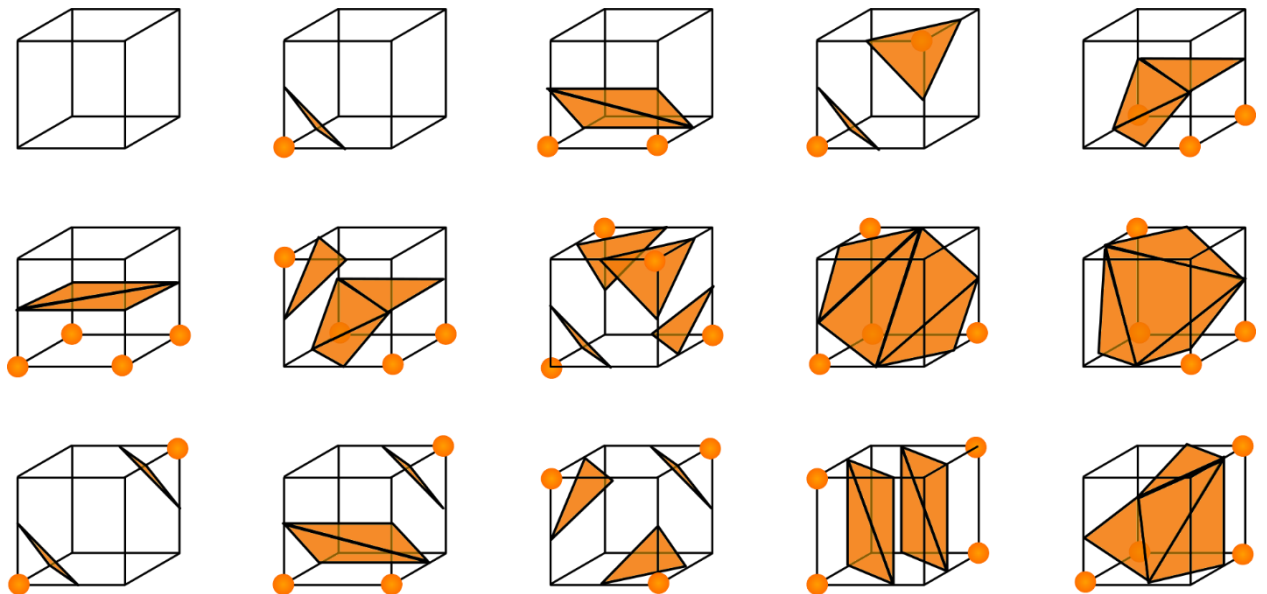


Рисунок 3.13 – 15 унікальних конфігурацій

4. Генерація поверхні: Використовуючи таблицю пошуку, алгоритм генерує сітку поверхні для кожної конфігурації куба. Таблиця пошуку містить список трикутників, які з'єднують вершини куба у спосіб, що апроксимує поверхню об'єкта або середовища. Кожен трикутник визначається трьома вершинами, які інтерполюються зі скалярних значень на ребрах куба.

5. Зшивання поверхні: Для створення остаточної сітки поверхні алгоритм зшиває разом сітки поверхонь усіх кубів у воксельній сітці. Процес зшивання видаляє всі трикутники, що повторюються, і з'єднує вершини сусідніх трикутників, утворюючи водонепроникну сітку.

6. Пост-обробка: Після того, як сіть поверхні згенеровано, її може знадобитися постобробка, щоб видалити будь-які артефакти, згладити межі або виокремити цікаві об'єкти.

Алгоритм Marching Cubes дуже добре налаштовується, і існує багато варіацій та розширень, які можуть покращити його продуктивність або точність. Наприклад, деякі варіанти використовують адаптивне розбиття, щоб зосередитися на ділянках з високою кривизною, тоді як інші використовують інформацію про градієнт для кращого наближення до нормалей поверхні. Загалом, Marching Cubes - це потужний алгоритм для генерації сіток поверхонь з об'ємних даних, який став стандартним інструментом у багатьох галузях 3D-візуалізації та моделювання.

Plenoxels - це метод перетворення карт глибини в 3D воксельні сітки, який був представлений в статті 2016 року M. Zollhöfer та ін. Метод Plenoxels призначений для створення високоякісного 3D представлення об'єкта або сцени на основі карти глибини з одного ракурсу, отриманої за допомогою Time-of-Flight (ToF) камери[30].

Нижче наведено детальний опис методу Plenoxels:

1. Вхідні дані: Метод Plenoxels приймає на вхід карту глибини одного ракурсу, яка являє собою двовимірне зображення, де кожен піксель

представляє відстань від камери до відповідної точки в сцені. Передбачається, що карта глибини вирівняна з кольоровим зображенням того ж виду.

2. Обробка: Метод Plenoxels спочатку обробляє карту глибини для видалення шуму, заповнення пропусків і згладжування з урахуванням країв. Ця обробка необхідна для того, щоб гарантувати, що результуюча воксельна сітка має високу якість і не містить артефактів.

3. Вокселізація: Метод Plenoxels перетворює оброблену карту глибини на 3D воксельну сітку, де кожен воксель представляє маленьку кубічну комірку в 3D просторі. Воксельна сітка вирівнюється з системою координат камери, а розмір кожного вокселя вибирається так, щоб збалансувати точність і роздільну здатність представлення.

4. Заповнення: Метод Plenoxels заповнює сітку вокселів, поширюючи значення глибини з вхідної карти глибини на вокселі. Цей процес заповнення виконується шляхом проектування значень глибини на площини вокселів та інтерполяції значень у межах кожного вокселя. Результатом є щільне 3D представлення сцени (рис. 3.14).

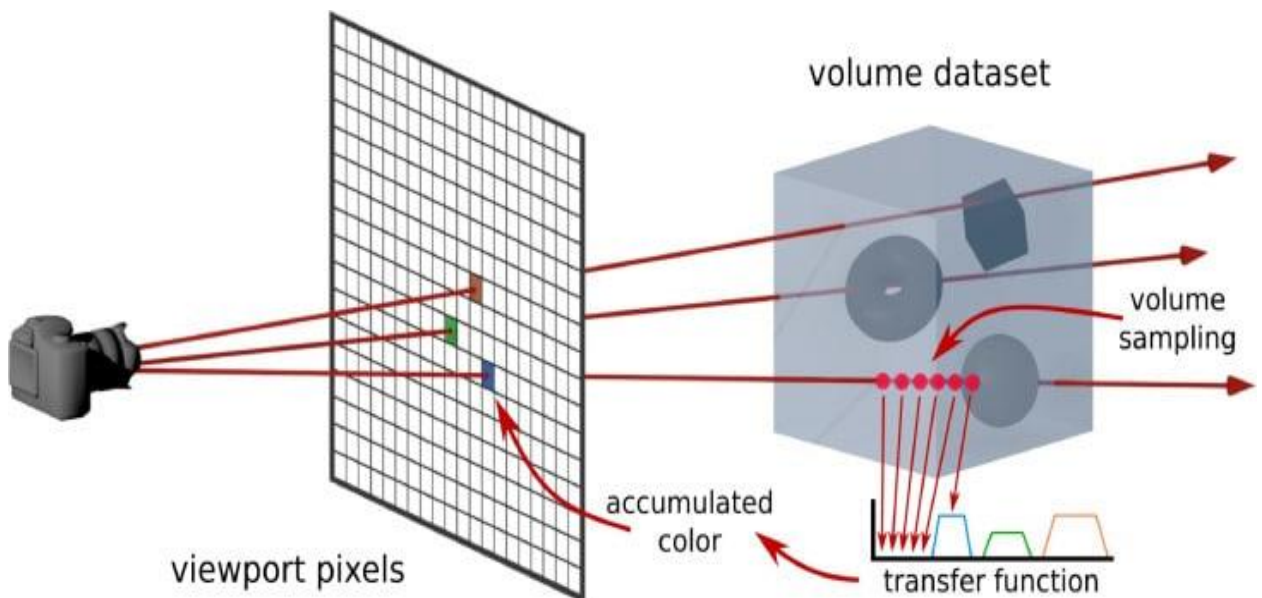


Рисунок 3.14 – Заповнення об'єму вокселів

5. Пост-обробка: Метод Plenoxels може виконувати постобробку воксельної сітки для подальшого покращення якості представлення. Наприклад, метод може застосовувати згладжування поверхні, уточнення поверхні або оцінку кольору на основі вхідного кольорового зображення.

Метод Plenoxels має кілька переваг над іншими методами вокселізації, наприклад, можливість обробляти зашумлені та неповні карти глибини, створювати високоякісні 3D-репрезентації та працювати з одним видом без необхідності використання декількох точок зору або складної геометрії з декількома видами. Метод Plenoxels використовується в різних сферах, таких як віртуальна і доповнена реальність, 3D-друк і робототехніка.

2.8 Розробка програмного забезпечення

Для початку необхідно встановити операційна система Raspbian на Raspberry Pi. Завантажуємо операційну систему Raspbian з офіційного сайту Raspberry Pi. Наявна можливість вибрати версію Raspbian Desktop або Raspbian Lite. Після того, як образ Raspbian завантажено, вставляємо карту microSD у комп'ютер і відформатуємо її. Можна скористатися інструментом форматування SD Card Association або вбудованим інструментом форматування комп'ютера. Для запису образу Raspbian на карту microSD можна використати програмою типу BalenaEtcher. Дотримуйтесь інструкцій, наданих програмою. Після того, як образ Raspbian буде записано на карту microSD, необхідно вставити її у Raspberry Pi. Далі Raspberry Pi підключається до монітора, клавіатури і миші та вмикається. Використовуючі інструкцій на екрані, щоб завершити процес налаштування. Необхідно дотримуватись інструкцій на екрані, щоб завершити процес налаштування. Після завершення налаштування вможна почати використовувати Raspbian на Raspberry Pi. Важливо зазначити, що операційна система Raspbian може потребувати оновлення або встановлення додаткового програмного забезпечення для сумісності з конкретною

конфігурацією обладнання або запланованими програмами. Крім того, на Raspberry Pi можна встановити інші операційні системи, такі як Ubuntu, Arch Linux та Windows 10 IoT Core.

Для початку роботи з датчиками необхідно встановити відповідні драйвери для сенсорного модуля MPU-6050 та камери ArduCam ToF на Raspberry Pi.

Для підключення сенсорного модуля MPU-6050 слід відкрити вікно терміналу на Raspberry Pi і ввести наступну команду для встановлення необхідних бібліотек (рис 3.15).

```
sudo apt-get install -y python-smbus i2c-tools
```

Рисунок 3.15 – Встановлення бібліотек для Raspberry Pi

Підключаємо модуль датчика MPU-6050 до Raspberry Pi за допомогою виводів I2C та введемо наступну команду у вікні терміналу, щоб виявити модуль датчика MPU-6050 (рис. 3. 16).

```
sudo i2cdetect -y 1
```

Рисунок 3.16 – Підключення MPU-6050 до Raspberry Pi

На виході повинна відобразитися адреса модуля датчика MPU-6050 (0x68 або 0x69).

Встановлюємо бібліотеку MPU-6050 Python, ввівши у вікні терміналу наступну команду (рис. 3.17).

```
sudo pip install mpu6050-raspberrypi
```

Рисунок 3.16 – Підключення MPU-6050 до MPU-6050

Сенсорний модуль MPU-6050 тепер готовий до використання з вашим Raspberry Pi.

Підключаємо камеру ArduCam ToF до Raspberry Pi за допомогою кабелю. У вікно терміналу на Raspberry Pi вводиться наступна команда для встановлення необхідного програмного забезпечення (рис 3. 17).

```
sudo apt-get install -y git
```

Рисунок 3.17 – Встановлюємо git

Далі вводимо команди для клонування драйверів камери ArduCam ToF зі сховища GitHub (рис. 3.18).

```
cd ~  
git clone https://github.com/ArduCAM/RaspberryPi.git  
cd RaspberryPi/TOF
```

Рисунок 3.18 – Клонування драйверів камери ArduCam

Компілюємо драйвери, ввівши наступні команди (рис. 3.19).

```
make  
sudo make install
```

Рисунок 3.19 – Компіляція драйверів

Перезавантажуємо Raspberry Pi, ввівши наступну команду (рис. 3.20).

```
sudo reboot
```

Рисунок 3.20 – Перезавантаження Raspberry Pi

Камера ArduCam ToF тепер готова до використання з Raspberry Pi. Тепер можна протестувати камеру, ввівши наступну команду (рис. 3.21).

```
tofcamtest
```

Рисунок 3.21 – Запуск тестування ToF камери

Якщо все зроблено правильно то можна буде спостерігати вивід даних.

```
fps:30  
select Rect distance: 1.34798  
select Rect distance: 1.33588  
select Rect distance: 1.35857  
select Rect distance: 1.35233  
select Rect distance: 1.34664
```

Рисунок 3.22 – Вивід даних з ToF камери

Далі необхідно встановити Plenoxels. Для цього з початку слід встановити необхідні залежності, ввівши команду зображену на рис. 3. 23.

```
sudo apt-get install -y libboost-all-dev cmake libpng-dev libeigen3-dev
```

Рисунок 3.23 – Встановлення необхідних залежностей

Клонуємо репозиторій Plenoxes з GitHub, ввівши наступну команду (рис. 3. 24).

```
git clone https://github.com/sxyu/svox2
```

Рисунок 3.24 – Клонування Plenoxes з GitHub

Перейдіть до каталогу plenoxes, ввівши наступну команду (рис. 3.25).

```
cd plenoxes
```

Рисунок 3.25 – Перейхід до каталогу plenoxes

Створюємо каталог збірки, ввівши наступну команду (рис. 3. 26).

```
mkdir build
```

Рисунок 3.26 – Створення каталогу збірки

Змінемо каталог збірки, ввівши наступну команду (рис. 3.27).

```
cd build
```

Рисунок 3.27 – Зміна каталогу збірки

Кафедра комп'ютерної інженерії
Апаратно-програмний комплекс для real-time симуляції руху об'єкту на базі
гіроскопічного датчика MPU-6050
Робимо налаштування збірки, ввівши наступну команду (рис. 3.28).

```
cmake ..
```

Рисунок 3.28 – Налаштування збірки

Збираємо Plenoxes, ввівши наступну команду (рис. 3.29)

```
make
```

Рисунок 3.29 – Збірка Plenoxes

Встановлюємо Plenoxes, ввівши наступну команду (рис. 3.30).

```
sudo make install
```

Рисунок 3.30 – Встановлення Plenoxes

Далі буде наведено код на Python для створення карти глибини з камери ToF і перетворення її в 3D воксельний об'єкт за допомогою Plenoxels на Raspberry Pi.

Для початку виконується імпорт бібліотек (рис. 3.31). Цей блок коду імпортує необхідні бібліотеки та модулі для скрипту. Ми імпортуємо cv2 для відображення та збереження зображень, numpy для маніпулювання даними зображення, Plenoptical для роботи з пленоптичною камерою та plenoxes для перетворення карти глибини у воксельний об'єкт.

```
import cv2
import numpy as np
from plenoptical import Plenoptical
from plenoxes import plenoxes
```

Рисунок 3.31 – Імпорт бібліотек

Ініціалізуйте пленоптичну камеру. На рис. 3.32. блок коду ініціалізує камеру ArduCam ToF за допомогою бібліотеки Plenopticalcam.

```
plenopticalcam = Plenopticalcam(camera_type='arducam')
```

Рисунок 3.31 – Ініціалізація пленоптичної камери

Встановлюємо параметри експозиції та балансу яскравості (рис. 3.32). Цей блок коду встановлює параметри експозиції та підсилення камери на відповідні значення. У цьому прикладі ми встановили час експозиції 1000 мікросекунд і коефіцієнт підсилення 1.0. Ці значення можна змінювати залежно від умов освітлення та сцени, що сканується.

```
plenopticalcam.set_exposure(1000)  
plenopticalcam.set_gain(1.0)
```

Рисунок 3.32 – Встановлення параметри експозиції та балансу яскравості

На рис. 3.33 блок коду записує карту глибини з камери ArduCam ToF.

```
depth_image = plenopticalcam.get_depthmap()
```

Рисунок 3.33 – Отримання карти глибини з камери ArduCam ToF

Далі створюємо бінарну маску. Цей блок коду створює бінарну маску із зображення глибини. Пікселі з ненульовими значеннями глибини отримують значення 255 (білий колір), а пікселі з нульовими значеннями глибини (тобто неприпустимі пікселі) отримують значення 0 (чорний колір).

```
mask = np.where(depth_image > 0, 255, 0).astype(np.uint8)
```

Рисунок 3.33 – Створення бінарної маски

Далі слід використати медіанну фільтрацію до зображення глибини для зменшення шуму (рис. 3.34).

```
depth_image = cv2.medianBlur(depth_image, 3)
```

Наступний блок коду перетворює карту глибини у воксельний об'єкт за допомогою Plenoxes. Параметр `voxel_size` задає розмір вокселів у метрах (рис. 3.35).

```
voxel_size = 0.01  
voxel_object = plenoxes.from_depth_map(depth_image, voxel_size)
```

Рисунок 3.35 – Перетворення карти глибини на воксельний об'єкт

Використовуючи функцію OpenCV `imshow`, цей блок коду перетворює воксельний об'єкт на зображення (рис. 3.36).

```
cv2.imshow("Voxel Object", voxel_object.view())
```

Рисунок 3.36 – Відображення воксельного об'єкта

Далі зберігаємо воксельний об'єкт у вигляді PLY-файлу (рис. 3.37).

```
voxel_object.save("voxel_object.ply")
```

Рисунок 3.37 – Збереження воксельного об'єкту

Ініціалізуємо код виходу який очікує на натискання клавіші і завершує роботу скрипта, коли клавіша натиснута (рис. 3.38).

```
cv2.waitKey(0)  
cv2.destroyAllWindows()
```

Рисунок 3.38 – Ініціалізація виходу

Висновки до розділу 3

В рамках дипломної роботи було проведено розробку системи real time симуляції руху об'єкту на базі гіроскопічного датчика MPU-6050. У результаті були розроблені такі елементи системи, як алгоритм визначення переміщення, концептуальна схема апаратно-програмного комплексу,

Кафедра комп'ютерної інженерії
Апаратно-програмний комплекс для real-time симуляції руху об'єкту на базі
гіроскопічного датчика MPU-6050
структурна діаграма схеми апаратно-програмного комплексу та діаграма
послідовності апаратно-програмного комплексу.

PA
GE
\
M
ER
GE
FO
R
M

ВИСНОВКИ

Дипломна робота була присвячена розробці апаратно-програмного комплексу на базі MPU-6050 та ToF-камери для створення віртуального середовища певної місцевості з елементами доповненої реальності у режимі реального часу для покращення навігації. Для досягнення цієї мети було виконано завдання, що полягали у проведенні аналізу апаратної частини, розробці програмного забезпечення, розробці алгоритму обробки отриманих даних та підборі необхідних компонентів.

Основною новизною роботи є створення апаратно-програмного комплексу, що базується на датчику MPU-6050 та ToF-камері для побудови віртуального середовища місцевості з елементами доповненої реальності в режимі реального часу з метою поліпшення навігації.

У першому розділі роботи були проаналізовані існуючі методи навігації та визначені вимоги до системи. Було обрано рішення про створення економічної та малоенергетичної інерційної системи навігації з використанням ToF-камери для динамічного створення оточення у 3D-просторі.

У другому розділі роботи були вибрані необхідні компоненти апаратно-програмного комплексу, такі як мікроконтролер Raspberry Pi 4 Model B, гіроскопічний датчик MPU-6050 та ToF-камера ArduCam. Також були розглянуті основні програмні засоби.

У третьому розділі було проведено розробку системи реалізації руху об'єкту в режимі реального часу на базі гіроскопічного датчика MPU-6050. Були розроблені алгоритм визначення переміщення, концептуальна схема апаратно-програмного комплексу, структурна діаграма схеми апаратно-програмного комплексу та діаграма послідовності апаратно-програмного комплексу.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. N. Thong-un, S. Hirata, and M. K. Kurosawa, “Three-dimensional-positioning based on echolocation using a simple iterative method,” *AEU - International Journal of Electronics and Communications*, vol. 69, no. 3, pp. 680–684, 2015.
2. P.-Y. Tseng, J. J. Lin, Y.-C. Chan, and A. Y. Chen, “Real-time indoor localization with visual SLAM for in-building emergency response,” *Autom Constr*, vol. 140, p. 104319, 2022.
3. E. Bonetto, P. Goldschmid, M. Pabst, M. J. Black, and A. Ahmad, “iRotate: Active Visual SLAM for Omnidirectional Robots,” *Rob Auton Syst*, vol. 154, p. 104102, 2022.
4. Q. Wang, C. Yan, R. Tan, Y. Feng, Y. Sun, and Y. Liu, “3D-CALI: Automatic calibration for camera and LiDAR using 3D checkerboard,” *Measurement*, vol. 203, p. 111971, 2022.
5. H. Yin, Z. Lin, and J. K. W. Yeoh, “Semantic localization on BIM-generated maps using a 3D LiDAR sensor,” *Autom Constr*, vol. 146, p. 104641, 2023.
6. X.-L. Yan et al., “Characterization of a double Time-Of-Flight detector system for accurate velocity measurement in a storage ring using laser beams,” *Nucl Instrum Methods Phys Res A*, vol. 931, pp. 52–59, 2019.
7. Z. Shang and Z. Shen, “Single-pass inline pipeline 3D reconstruction using depth camera array,” *Autom Constr*, vol. 138, p. 104231, 2022.
8. Y. Liu, F. Zhou, Z. Guo, H. Tan, and W. Zhang, “Design and optimization of a quad-directional stereo vision sensor with wide field of view based on single camera,” *Measurement*, vol. 203, p. 111915, 2022.
9. S. Mansoor, U. I. Bhatti, A. I. Bhatti, and S. M. D. Ali, “Improved attitude determination by compensation of gyroscopic drift by use of accelerometers and magnetometers,” *Measurement*, vol. 131, pp. 582–589, 2019.
10. A.-S. Haider Ali Muse, “Principles of constructing gyroscopes based on photonic crystal (band-gap) fibers,” *Radiotekhnika*, no. 205, pp. 100–107, Jul. 2021.

11. What is an Arduino? URL: <https://learn.sparkfun.com/tutorials/what-is-an-arduino/all>
12. Arduino Nano. URL: <https://store.arduino.cc/products/arduino-nano>
13. How to Program Arduino Nano. URL: <https://linuxhint.com/program-arduino-nano/>
14. Finding Raspberry Pi: Where to buy the latest model and its alternatives. URL: <https://www.zdnet.com/article/finding-raspberry-pi-where-to-buy-the-latest-model-and-its-alternatives/>
15. Raspberry Pi 4 Model B review. URL: <https://www.tomsguide.com/reviews/raspberry-pi-4-model-b>
16. MPU6050 (Gyroscope + Accelerometer + Temperature) Sensor Module. URL: <https://www.electronicwings.com/sensors-modules/mpu6050-gyroscope-accelerometer-temperature-sensor-module>
17. Introduction to MPU6050. URL: <https://www.theengineeringprojects.com/2019/02/introduction-to-mpu6050.html>
18. Three-Axis Digital Compass IC HMC5883L. URL: <https://www.farnell.com/datasheets/1683374.pdf>
19. WitMotion WT61 6 Axis AHRS Sensor Digital Tilt Angle Inclinator + Accelerometer + Gyroscope. URL: <https://www.wit-motion.com/6-axis/witmotion-wt61-6-axis-ahrs-sensor-digital.html>
20. Time of Flight (ToF) Camera for Raspberry Pi – Arducam. URL: <https://www.wit-motion.com/6-axis/witmotion-wt61-6-axis-ahrs-sensor-digital.html>
21. Arducam Launches Active Time-of-Flight Depth Camera Module for the Raspberry Pi. URL: <https://www.hackster.io/news/arducam-launches-active-time-of-flight-depth-camera-module-for-the-raspberry-pi-c5453e68514a>
22. New ArduCam Time of Flight (ToF) Camera - The DIY Life. URL: <https://www.the-diy-life.com/new-arducam-time-of-flight-tof-camera-your-pi-can-now-see-in-3d/>

23. Time of Flight (ToF) Camera for Raspberry Pi by Arducam. URL:
<https://www.kickstarter.com/projects/arducam/time-of-flight-tof-camera-for-raspberry-pi>
24. МОДУЛЬ ВІДСТАНІ VL53L1X 4M WAVESHARE. URL:
<https://miniboard.com.ua/sensors/780-modul-rasstoyaniya-vl53l1x-4m-waveshare.html>
25. Waveshare 14929 VL53L1X Distance Sensor. URL:
https://www.welectron.com/Waveshare-14929-VL53L1X-Distance-Sensor_1
26. Introduction to ZeroMQ. URL: <https://se-education.org/learningresources/contents/zeromq/zeromq.html>
27. Amazon EC2 G5 Instances. URL: <https://aws.amazon.com/ec2/instance-types/g5/>
28. Amazon DynamoDB. URL: <https://aws.amazon.com/dynamodb/>
29. Marching Cubes Part 1: Explaining marching cubes. URL:
<https://polycoding.net/marching-cubes/part-1/#understanding>
30. Plenoxels: Radiance Fields without Neural Networks. URL:
<https://alexyu.net/plenoxels/>
31. Чернявський Р. А., Пузирьов С. В. Орієнтація у просторі на базі гіроскопічного датчика MPU-6050. Могилянські читання – 2022 : тези доп. XXV Всеукр. наук.-метод. конф. Миколаїв, 7–11 листоп. 2022 р. Миколаїв : Чорном. нац. ун-т ім. Петра Могили, 2022. С. 101–104.

ДОДАТОК А

**КОНЦЕПТУАЛЬНА СХЕМА АПАРАТНО-ПРОГРАМНОГО
КОМПЛЕКСУ**

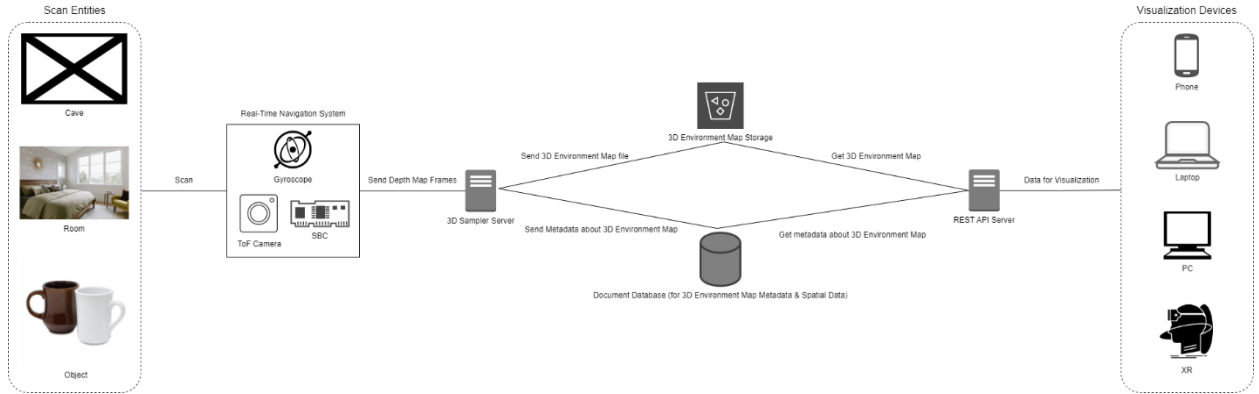


Рисунок А.1 – Концептуальна схема апаратно-програмного комплексу

ДОДАТОК Б

**СТРУКТУРНА ДІАГРАМА АПАРАТНО-ПРОГРАМНОГО
КОМПЛЕКСУ**

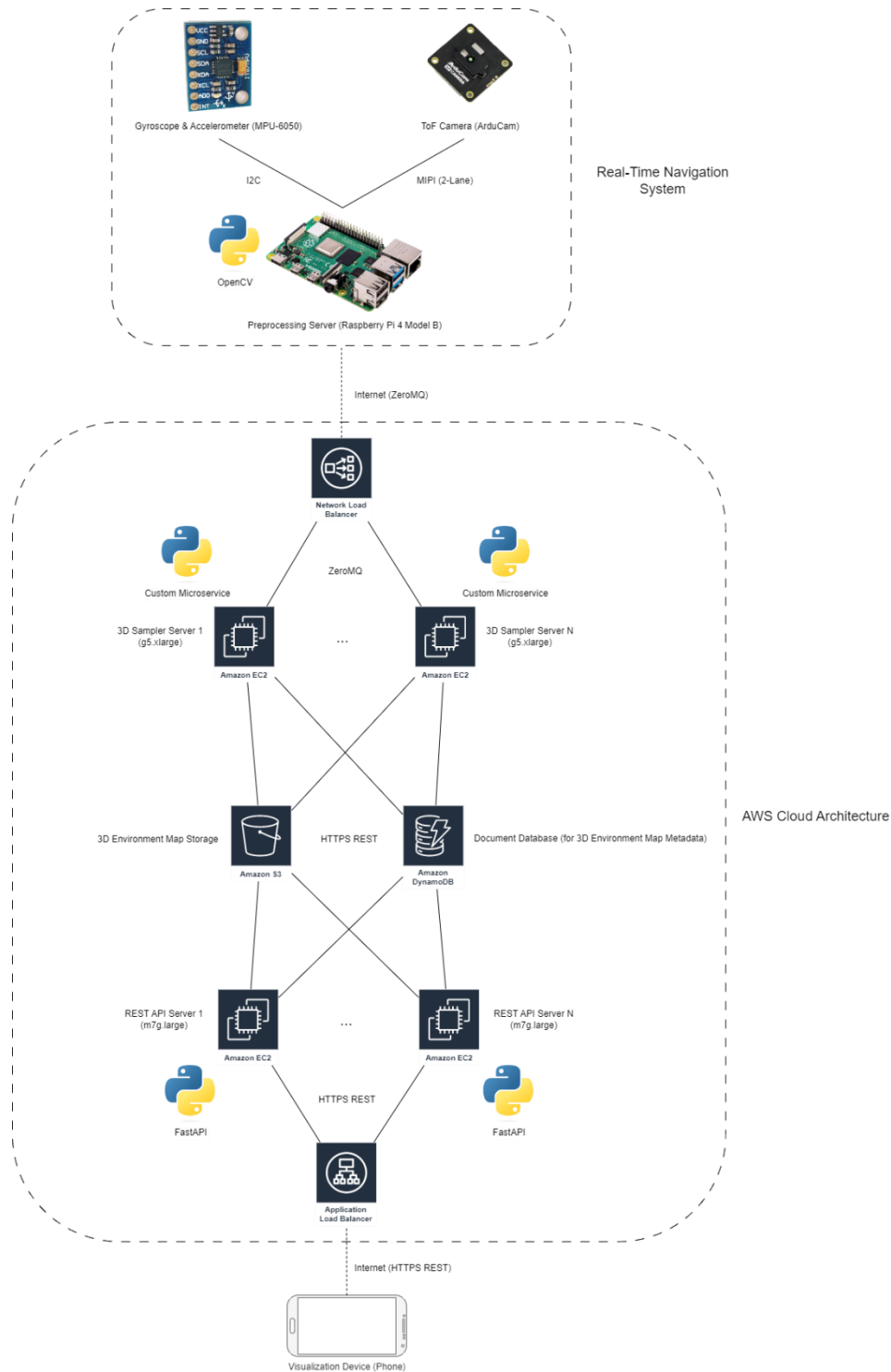


Рисунок Б.1 – Діаграма апаратної архітектури апаратно-програмного
КОМПЛЕКСУ

ДОДАТОК Г

ПРОГРАМНИЙ КОД АПАРАТНО-ПРОГРАМНОГО КОМПЛЕКСУ

```
import time
import numpy as np
from plenopticalcam import Plenopticalcam
from plenoxes import plenoxes

# Initialize the plenoptic camera
plenopticalcam = Plenopticalcam(camera_type='arducam')

# Set the exposure and gain parameters
plenopticalcam.set_exposure(1000) # set the exposure time to 1000 microseconds
plenopticalcam.set_gain(1.0) # set the gain to 1.0

# Capture a depth map from the ArduCam ToF camera
depth_image = plenopticalcam.get_depthmap()

# Create a binary mask from the depth image
mask = np.where(depth_image > 0, 255, 0).astype(np.uint8)

# Apply median filtering to the depth image to reduce noise
depth_image = cv2.medianBlur(depth_image, 3)

# Convert the depth map to a voxel object using Plenoxes
voxel_size = 0.01 # set the size of the voxels in meters
voxel_object = plenoxes.from_depth_map(depth_image, voxel_size)

# Display the voxel object
cv2.imshow("Voxel Object", voxel_object.view())

# Save the voxel object as a PLY file
voxel_object.save("voxel_object.ply")

# Wait for a key press and exit
cv2.waitKey(0)
cv2.destroyAllWindows()
```

ДОДАТОК Г

ДОВІДКА

про перевірку на унікальність пояснювальної записки кваліфікаційної магістерської
роботи

на тему: «Real time симуляція руху об'єкту на базі гіроскопічного датчика MPU-6050»
студента спеціальності 123 «Комп'ютерна інженерія», групи 605м

Чернявський Руслан Андрійович

прізвище, ім'я, по-батькові

Перевірку тексту здійснено сервісом: онлайн-сервіс

Unicheck. Результат перевірки тексту роботи:

схожість складає 1,4%.



User name:
Сергій Пузирьов

Check ID:
1014060905

Check date:
19.02.2023 17:54:23 EET

Check type:
Doc vs Internet + Library

Report date:
19.02.2023 18:00:38 EET

User ID:
100000135

File name: **605_Чернявський_MP_2023**

Page count: **10** Word count: **10423** Character count: **76579** File size: **90.51 KB** File ID: **1013800951**

1.4% Matches

Highest match: **0.79%** with Library source (File ID: **1013774011**)

0.11% Internet sources 1 Page 12

1.3% Library sources 23 Page 12

0% Quotes

Exclusion of quotes is off

Exclusion of references is off

0% Exclusions

No exclusions

Modifind

Text modifications detected. Find more details in the online report.

Replaced characters 6

Студент

_____ Р. А. Чернявський
підпис ініціали, прізвище

Дата: «__» _____ 2023 р.

Керівник

канд. фіз.-мат. наук, доцент

_____ С. В. Пузирьов
підпис ініціали, прізвище