

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: *Комп'ютерна система відтворення рухів людини у віртуальному просторі на базі гіроскопів*

Виконав(ла): студент(ка) *IV* курсу, групи *СІЗс-42*

спеціальності *123 «Комп'ютерна інженерія»*

(шифр і назва спеціальності)

(підпис)

Соколов О. М.

(прізвище та ініціали)

Керівник

(підпис)

Жаровський Р. О.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Тих С. В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Осухівська Г. М.

(прізвище та ініціали)

Рецензент

(підпис)

Никитюк В. В.

(прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Осухівська Г.М.

(підпис)

(прізвище та ініціали)

« ____ » _____ 2023 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня бакалавр
(назва освітнього ступеня)

за спеціальністю 123 «Комп'ютерна інженерія»
(шифр і назва спеціальності)

студенту Соколов Олександр Михайлович
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютерна система відтворення рухів людини у віртуальному просторі на базі гіроскопів

Керівник роботи Жаровський Руслан Олегович, к.т.н., ст.викл
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 16 » 05 2023 року № 417-556

2. Термін подання студентом завершеної роботи 14.06.2023р.

3. Вихідні дані до роботи Проектування VR трекера

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Аналіз технічного завдання

2. Проектна частина

3. Практична частина

4. Безпека життєдіяльності, основи охорони праці.

Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Структура системи захоплення рухів VR

2. Алгоритм роботи плагіну для Steam VR

3. Схема VR трекера на базі BNO85

4. Скріншоти з результатами роботи VR трекерів

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Безпека життєдіяльності, основи охорони праці</i>	Пилипець М. І д-р. техн. наук, професор		

7. Дата видачі завдання 17.05 2023

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	<i>Розробка технічного завдання</i>	<i>17.05-20.05.2033</i>	
2	<i>Аналіз технічного завдання</i>	<i>22.05-24.05.2033</i>	
3	<i>Аналіз вимог до система відтворення рухів людини у VR</i>	<i>25.05-26.05.2033</i>	
4	<i>Розробка програмного забезпечення</i>	<i>01.06-04.06.2033</i>	
5	<i>Розробка схем VR трекерів</i>	<i>05.06-07.06.2033</i>	
6	<i>Перевірка роботи VR трекерів</i>	<i>08.05-10.06.2033</i>	
7	<i>Безпека життєдіяльності, основи охорони праці</i>	<i>11.06-12.06.2033</i>	
8	<i>Оформлення кваліфікаційної роботи</i>	<i>12.06-13.06.2033</i>	
9	<i>Захист кваліфікаційної роботи</i>	<i>14.06.2033</i>	

Студент

(підпис)

Соколов Олександр Михайлович

(прізвище та ініціали)

Керівник роботи

(підпис)

Жаровський Руслан Олегович

(прізвище та ініціали)

АНОТАЦІЯ

Комп'ютерна система відтворення рухів людини у віртуальному просторі на базі гіроскопів // Кваліфікаційна робота на здобуття освітнього ступеня бакалавр // Соколов Олександр Михайлович // ТНТУ, спеціальність 123 «Комп'ютерна інженерія» // СІзс-42 // Тернопіль. 2023 рік // с. – 66, рис –37, плакати – 4, додат. – 9, бібліогр. – 17.

Ключові слова: VR трекер, Full body, трекінг, плагін, Steam Vr.

В курсовому проєкті побудовано систему захоплення і передачі рухів людини в віртуальний простір за допомогою використання VR трекерів. Описано загальну структуру та принцип роботи VR трекерів. Наведено опис деталей трекера, алгоритм роботи плагіну для Steam VR, та структури системи. Описано використовувані команди та код плагіну для Steam VR.

Пояснювальна записка містить 4 розділи.

В першому розділі здійснюється аналіз предметної області. Наведено пояснення що таке VR трекер та види трекінгу. Докладно розглянуто види трекінгу їх переваги та недоліки.

В другому розділі описані компоненти і особливості побудови VR трекера та наведено алгоритм роботи плагіну для програми Steam VR.

В третьому розділі показано як прошити компоненти VR трекера, підключення трекера до ПК та результати роботи VR трекера.

В четвертому розділі розглянуті питання охорони праці та вимоги з техніки безпеки в VR.

ABSTRACTS

Computer system for reproduction of human movements in virtual space based on gyroscopes // Qualification work for the degree of Bachelor // Sokolov Oleksandr Mykhailovych // TNTU, specialty 123 "Computer Engineering" // CIzs-42 // Ternopil. 2023 // p. – 66, fig -37, posters – 4, appendix – 9, bibliogr. – 17.

Key words: VR tracker, Full body, tracking, plugin, Steam VR.

In the course project, a system for capturing and transmitting human movements into virtual space using VR trackers is built. The general structure and principle of operation of VR trackers are described. A description of the tracker details, the algorithm of the Steam VR plugin, and the system structure is given. The commands used and the code of the Steam VR plugin are described.

The explanatory note contains 4 sections.

The first section analyzes the subject area. It explains what a VR tracker is and the types of tracking. The types of tracking, their advantages and disadvantages are discussed in detail.

The second section describes the components and features of building a VR tracker and provides the algorithm of the plugin for the Steam VR program.

The third section shows how to flash VR tracker components, connect the tracker to a PC, and the results of VR tracker operation.

The fourth section discusses labor protection issues and safety requirements in VR.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ.....	9
1.1 Аналіз вимог до системи.	9
1.2 Аналіз предметної області.	9
1.2.1 VR трекер.	9
1.2.2 Трекери які відслідковуються за допомогою RGBD-камер	9
1.2.3 Full-body tracking/ або які є види відслідковування трекерів	10
1.2.4 Оптичне відстеження	12
1.2.5 Відстеження ззовні всередину	13
1.2.6 Відстеження навиворіт	14
1.2.7 Інерційне відстеження	15
1.2.8 Магнітне відстеження.....	17
1.3 Вимоги до системи відтворення руху людини у віртуальному просторі..	18
.....	18
РОЗДІЛ 2 ПРОЕКТНА ЧАСТИНА	19
2.1 Структура системи захоплення рухів в VR.....	19
2.2 Розробка плану для Steam VR	19
2.3 Передача даних за допомогою Wi-Fi	21
2.4 Калібрування трекерів у VR.....	24
2.5 Апаратна реалізація	24
2.5.1 Варіанти схем для VR трекера.....	24
2.5.2 Зарядний модуль TP4056 Type-C	30
2.5.3 Wi-Fi модуль ESP8266 WeMos D1 Mini	32
РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА	37
3.1 Програмна реалізація.....	37

					КС КРБ 123.023.00.00 ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Комп'ютерна система відтворення рухів людини у віртуальному просторі на базі гіроскопів	Літ.	Арк..	Акрушів
Розроб.		Соколов О.М.					6	57
Перевір.		Жаровський Р.О						
Рецез.		Никитюк В. В				ТНТУ, каф. КС, гр. СІзс-		
Н. контр.		Тиш С. В				42		
Затв.		Осухівська Г.М.						

3.1.1 Встановлення бібліотеки та прошивка BNO085	37
3.1.2 Прошивка коду на ESP8266 WeMos D1 Mini.....	39
3.2 Об'єднання та підключення трекера(ів) в одну систему	40
3.3 Перевірка працездатності системи Full Body у віар просторі.	43
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ.	48
4.1 Техніка безпеки експлуатації гарнітури віртуальної реальності.....	48
4.2 Долікарська допомога при контузіях.....	42
ВИСНОВКИ.....	55
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	56
Додаток А Технічне завдання	58
Додаток Б Лістинг програмного забезпечення	64

					КС КРБ 123.023.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

ВСТУП

Віртуальна реальність (VR) має великий потенціал у багатьох сферах, включаючи соціальні мережі, ігри та розваги. Однак сучасні широко розповсюджені системи віртуальної реальності можуть відстежувати лише положення голови та рук користувача, але не решти його тіла. Тому більшість VR-додатків сьогодні або відображають лише верхню частину тіла користувача, або передбачають положення нижньої частини тіла користувача відповідно до його рухів верхньої частини тіла. В результаті плаваючі аватари і несинхронізовані рухи ніг можуть зруйнувати ілюзію користувача, перешкоджати самовираженню користувача і обмежувати типи додатків, які можуть створювати розробники.

Однак сучасні рішення для відстеження всього тіла мають різні обмеження. Сенсорне відстеження або відстеження ззовні може давати точні результати, але користувачам потрібно або носити громіздкі 3D-позиційні трекери по всьому тілу, або використовувати RGBD-камери, які потребують тривалого калібрування і які складно отримати (відстеження ззовні). З іншого боку, останні комерційні системи відстеження верхньої частини тіла VR покладаються на егоцентричні камери відстеження (так зване відстеження "зсередини назовні"), які вимагають мінімального налаштування, значно знижують бар'єр входу у віртуальну реальність і, як правило, користувачі надають їм перевагу. Однак дослідження показали, що егоцентричні камери не можуть забезпечити адекватне відстеження нижньої частини тіла через внутрішньотілесну оклюзію.

					КС КРБ 123.023.00.00 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Аналіз вимог до системи

Об'єктом розробки є система для передачі рухів людини у віар простір за допомогою VR трекерів які б в свою чергу працювали без допоміжних базових станцій (RGBD камер) на базі сенсорів гіроскопу та акселерометру.

Основним об'єктом розробки є самі VR трекери, потрібно буде розробити трекер який мають бути дешевшими відносно сучасних аналогів, за допомогою доступних та не дорогих деталей які легко можна знайти та замовити в Інтернеті.

1.2 Аналіз предметної області.

1.2.1 VR Трекер

Це невеликий пристрій, який кріпиться на різні частини тіла, в більногсті на таз та ноги. Такий пристрій служить для того, щоб "перенести" будь-який предмет або рух тіла у віртуальну реальність. Якщо бути точним, то перенести його рухи і положення в просторі. Vr трекери це окремі пристрої які працюють разом в одні системі за допомогою ПЗ, та використовуються для захоплення та передачі рухів людини у віртуальний простір. Трекери мають кілька видів.

1.2.2 Трекери які відслідковуються за допомогою RGBD-камер (Vive Tracker 1,2,3)

Такі трекери покладаються на базові станції/ RGBD-камери, які є зовнішніми пристроями, що випромінюють інфрачервоні сигнали. Ці базові станції створюють сітку відстеження в межах ігрової зони і випромінюють сигнали, які приймаються датчиками на трекері.

					КС КРБ 123.023.00.00 ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Соколов О.М.			Аналіз технічного завдання	Літ.	Арк..	Акрушів
Перевір.		Жаровський Р.О					9	55
Рецез.		Никитюк В. В				ТНТУ, каф. КС, гр. СІзс-42		
Н. контр.		Тиш С. В						
Затв.		Осухівська Г.М.						

Отримуючи ці сигнали, трекер може точно визначити своє положення і орієнтацію відносно базових станцій. Після того, як трекер фіксує дані з датчиків і отримує сигнали від базових станцій, він надсилає цю інформацію бездротовим зв'язком до системи віртуальної реальності, де вона обробляється. VR-система поєднує дані з трекера з віртуальним середовищем, дозволяючи користувачеві взаємодіяти з фізичними об'єктами у віртуальному просторі. Дані про положення і переміщення з трекера використовуються для точного відстеження фізичного об'єкта, до якого він прикріплений, у віртуальному середовищі. Це дозволяє користувачам бачити і маніпулювати віртуальними представленнями цих об'єктів так, ніби вони взаємодіють з ними в реальному світі.

Також є надзвичайно дорогі але дуже технологічні трекери які відслідковують самі себе за допомогою камер які і встановлені на них же. (розробляється компанією Vive та наразі немає точної назви тому як станом на 2023р він ще не вийшов у продаж)

1.2.3 Full-body tracking/ або які є види відслідковування трекерів

У віртуальній реальності (VR) і доповненій реальності (AR) система відстеження пози визначає точне положення наголовних дисплеїв, контролерів, інших об'єктів або частин тіла в евклідовому просторі. Відстеження пози часто називають 6DOF-відстеженням, за шістьма ступенями свободи, в яких часто відстежується поза. Відстеження пози іноді називають позиційним відстеженням, але ці два поняття є окремими. Відстеження пози відрізняється від позиційного відстеження тим, що відстеження пози включає орієнтацію, тоді як позиційне відстеження не включає. У деяких побутових GPS-системах дані про орієнтацію додаються додатково за допомогою магнітометрів, які дають часткову інформацію про орієнтацію, але не повну орієнтацію, яку забезпечує відстеження пози.

У VR дуже важливо, щоб відстеження пози було точним і акуратним, щоб не порушити ілюзію перебування у віртуальному світі.

					КС КРБ 123.023.00.00 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

Для цього було розроблено кілька методів відстеження положення та орієнтації (нахилу, повороту та обертання) дисплея та будь-яких пов'язаних з ним об'єктів чи пристроїв. Багато методів використовують датчики, які багаторазово записують сигнали від передавачів на об'єкті (об'єктах), що відстежуються, або поблизу нього (них), а потім надсилають ці дані на комп'ютер, щоб підтримувати наближення до їхнього фізичного місцезнаходження. Популярним методом відстеження є відстеження за допомогою датчиків. Загалом, ці фізичні місцезнаходження ідентифікуються та визначаються за допомогою однієї або декількох з трьох систем координат: декартової прямолінійної системи, сферичної полярної системи та циліндричної системи. Багато інтерфейсів також було розроблено для моніторингу та контролю переміщення у віртуальному 3D-просторі та взаємодії з ним; такі інтерфейси повинні тісно співпрацювати з системами відстеження положення, щоб забезпечити безперебійну роботу користувача.

Інший тип відстеження положення, який частіше використовується в новітніх системах, називається відстеженням "навиворіт", включаючи одночасну локалізацію і картографування/Simultaneous localization and mapping (SLAM) або візуально-інтервальну одометрію/Visual-inertial odometry (VIO). Прикладом пристрою, який використовує відстеження пози навиворіт, є Oculus Quest 2 (рис. 1.1) [6].

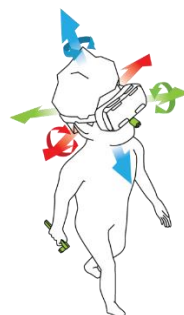


Рисунок 1.1 — Приклад (SLAM) та (VIO)

					КС КРБ 123.023.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

1.2.4 Оптичне відстеження

Оптичне відстеження використовує камери, розміщені на гарнітурі або навколо неї, для визначення положення та орієнтації на основі алгоритмів комп'ютерного зору. Цей метод заснований на тому ж принципі, що і стереоскопічний людський зір. Коли людина дивиться на об'єкт за допомогою біноклярного зору, вона може приблизно визначити, на якій відстані знаходиться об'єкт, завдяки різниці в перспективі між двома очима. В оптичному відстеженні камери калібруються для визначення відстані до об'єкта і його положення в просторі. Оптичні системи надійні і відносно недорогі, але їх може бути важко відкалібрувати. Крім того, системі потрібен прямий промінь світла без перешкод, інакше вона отримуватиме невірні дані.

Оптичне відстеження може здійснюватися як з маркерами, так і без них. Відстеження з маркерами передбачає наявність цілей з відомими шаблонами, які слугують орієнтирами, і камери постійно шукають ці маркери, а потім використовують різні алгоритми (наприклад, алгоритм POSIT) для визначення положення об'єкта. Маркери можуть бути видимими, наприклад, надруковані QR-коди, але багато хто використовує інфрачервоне (ІЧ) світло, яке можуть вловлювати лише камери. Активні реалізації мають маркери з вбудованими ІЧ-світлодіодами, які можуть вмикатися і вимикатися для синхронізації з камерою, що полегшує блокування інших ІЧ-світлодіодів у зоні відстеження. Пасивні реалізації - це світловідбивачі, які відбивають ІЧ-світло назад до джерела з невеликим розсіюванням. Відстеження без маркерів не потребує попередньо встановлених цілей, натомість використовує природні особливості навколишнього середовища для визначення положення та орієнтації (рис.1.2) [4].

					КС КРБ 123.023.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

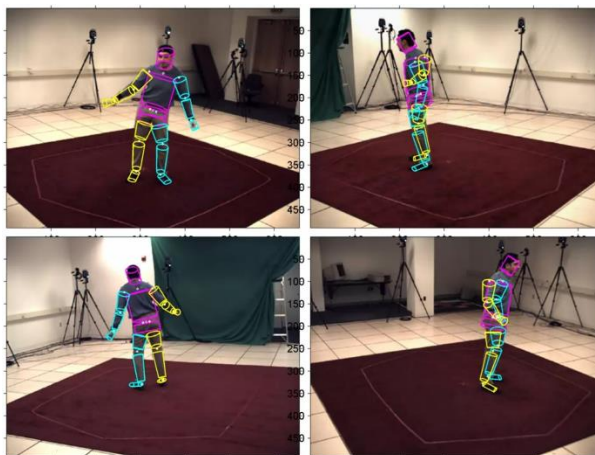


Рисунок 1.2 — Приклад оптичного відстежування.

1.2.5 Відстеження ззовні всередину

У цьому методі камери розміщуються в стаціонарних місцях у навколишньому середовищі, щоб відстежувати положення маркерів на відстежуваному пристрої, наприклад, на дисплеї або контролерах, що встановлюються на голові. Наявність декількох камер дозволяє отримати різні види одних і тих же маркерів, і це перекриття дозволяє точно зчитувати положення пристрою. Оригінальний Oculus Rift використовує цю техніку, розміщуючи сузір'я інфрачервоних світлодіодів на гарнітурі і контролерах, щоб дозволити зовнішнім камерам в навколишньому середовищі зчитувати їх положення. Цей метод є найбільш зрілим і застосовується не тільки в VR, але і в технології захоплення руху для кіно. Однак, це рішення обмежене в просторі і потребує постійного спостереження за пристроєм з боку зовнішніх датчиків.[1]

Переваги:

— Більш точні показники, які можна покращити, додавши більше камер.

— Менша затримка, ніж при відстеженні "зсередини назовні".

Недоліки:

— Ціна.

— Оклюзія, камери повинні знаходитися в зоні прямої видимості, інакше відстеження не працюватиме.

					КС КРБ 123.023.00.00 ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

— Необхідність зовнішніх датчиків означає обмежену площу ігрового простору.

1.2.6 Відстеження навиворіт

У цьому методі камера розміщується на пристрої, що відстежується, і дивиться назовні, щоб визначити його місцезнаходження в навколишньому середовищі. Гарнітури, які використовують цю технологію, мають кілька камер, спрямованих у різні боки, щоб отримати огляд усього оточення. Цей метод може працювати з маркерами або без них. Система Lighthouse, що використовується в HTC Vive, є прикладом активних маркерів. Кожен зовнішній модуль Lighthouse містить інфрачервоні світлодіоди, а також лазерний масив, який розгортається в горизонтальному і вертикальному напрямках, а датчики на гарнітурі і контролерах можуть виявляти ці розгортання і використовувати таймінги для визначення положення. Відстеження без маркерів, як, наприклад, в Oculus Quest, не вимагає нічого встановленого в зовнішньому середовищі. Він використовує камери на гарнітурі для процесу, який називається SLAM, або одночасна локалізація та картографування, де в режимі реального часу створюється 3D-карта навколишнього середовища, Алгоритми машинного навчання потім визначають, де гарнітура розташована на цій 3D-карті, використовуючи виявлення об'єктів для реконструкції та аналізу її оточення [3].

Ця технологія дозволяє висококласним гарнітурам, таким як Microsoft HoloLens, бути автономними, але також відкриває двері для дешевших мобільних гарнітур, які не потребують зв'язку з зовнішніми комп'ютерами або датчиками.

Переваги:

- Забезпечує більший ігровий простір, може розширюватися, щоб відповідати кімнаті.
- Адаптується до нових умов.

					КС КРБ 123.023.00.00 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

Недоліки:

- Потрібно більше вбудованої обробки.
- Затримка може бути вищою.

1.2.7 Інерційне відстеження

Інерційне відстеження використовує дані з акселерометрів і гіроскопів, а іноді й магнітометрів. Акселерометри вимірюють лінійне прискорення. Оскільки похідна положення за часом є швидкістю, а похідна швидкості є прискоренням, вихідні дані акселерометра можна інтегрувати, щоб знайти швидкість, а потім знову інтегрувати, щоб знайти положення відносно деякої початкової точки. Гіроскопи вимірюють кутову швидкість. Кутову швидкість можна також інтегрувати, щоб визначити кутову позицію відносно початкової точки. Магнітометри вимірюють магнітні поля і магнітні дипольні моменти. Напрямок магнітного поля Землі можна врахувати, щоб мати абсолютний орієнтир орієнтації і компенсувати гіроскопічні дрейфи. Сучасні інерціальні вимірювальні системи (IMU) базуються на технології MEMS, що дозволяє відстежувати орієнтацію (крен, тангаж, рискання) в просторі з високою швидкістю оновлення і мінімальною затримкою.

Гіроскопи завжди використовуються для відстеження обертання, але для позиційного відстеження застосовуються різні технології, які залежать від таких факторів, як вартість, простота налаштування та обсяг відстеження.

Статичний розрахунок використовується для відстеження позиційних даних, який змінює віртуальне середовище, оновлюючи зміни руху користувача.

Швидкість оновлення статичного розрахунку та алгоритм прогнозування, що використовуються в системі віртуальної реальності, впливають на досвід користувача, але немає консенсусу щодо найкращих практик, оскільки використовується багато різних методик. Важко покладатися лише на інерційне відстеження для визначення точної позиції, оскільки мертва точка призводить до дрейфу, тому цей тип відстеження не використовується ізольовано у віртуальній

					КС КРБ 123.023.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

–реальності. Було виявлено, що затримка між рухом користувача і відображенням віртуальної реальності понад 100 мс викликає нудоту.

Інерційні датчики здатні відстежувати не тільки обертальний рух (крен, тангаж, рискання), але й поступальний. Ці два типи руху разом відомі як шість ступенів свободи. Багато додатків віртуальної реальності повинні не тільки відстежувати обертання голови користувача, але й те, як рухається його тіло (вліво/вправо, вперед/назад, вгору/вниз). Шість ступенів свободи не є необхідними для всіх видів віртуальної реальності, але вони корисні, коли користувачеві потрібно рухати не тільки головою, але й іншими частинами тіла.

Переваги:

— Може добре відстежувати швидкі рухи відносно інших сенсорів, і особливо добре в поєднанні з іншими сенсорами.

— Здатний до високих частот оновлення.

— Відносно низька ціна.

Недоліки:

— Будь-яка затримка або прорахунки при визначенні положення можуть призвести до таких симптомів у користувача, як нудота або головний біль.

— Можуть не встигати за користувачем, який рухається занадто швидко.

— Інерційні датчики, як правило, можуть використовуватися лише в приміщеннях та лабораторіях, тому їх застосування на відкритому повітрі обмежене.[5]

1.2.8 Магнітне відстеження

Магнітний трекінг (або електромагнітний трекінг) базується на тому ж принципі, що і теремін. Воно покладається на вимірювання інтенсивності неоднорідних магнітних полів за допомогою електромагнітних датчиків. Базова станція, яку часто називають передавачем або генератором поля системи,

					КС КРБ 123.023.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

–генерує змінне або статичне електромагнітне поле, залежно від архітектури системи.

Щоб покрити всі напрямки в тривимірному просторі, три магнітні поля генеруються послідовно. Магнітні поля генеруються трьома електромагнітними котушками, які перпендикулярні одна до одної. Ці котушки повинні бути поміщені в невеликий корпус, встановлений на рухомій мішені, положення якої необхідно відстежувати. Струм, послідовно проходячи через котушки, перетворює їх на електромагніти, що дозволяє визначити їхнє положення та орієнтацію в просторі.

Оскільки магнітному відстеженню не потрібно дисплеїв на голові, які можна доволі часто побачити у старих системах віртуальної реальності. Звичайне обладнання, таке як наголовні дисплеї, є нав'язливим для користувача в повністю закритій віртуальній реальності, тому перевага надається альтернативному обладнанню, такому як те, що використовується в магнітному відстежуванні. Магнітне відстеження було реалізовано компанією Polhemus і в Razor Hydra компанією Sixense. Система погано працює поблизу будь-яких електропровідних матеріалів, таких як металеві предмети і пристрої, які можуть впливати на електромагнітне поле.

Магнітне відстеження погіршується, коли користувач віддаляється від базового випромінювача, а масштабована зона обмежена і не може бути більшою за 5 метрів.

Переваги:

— Використовує ненав'язливе обладнання, яке не потрібно носити користувачеві і яке не заважає користувачеві відчувати віртуальну реальність.

— Підходить для дисплеїв віртуальної реальності з повним зануренням.

Недоліки:

— Користувач має бути близько до базового випромінювача.

— Відстеження погіршується поблизу металів або об'єктів, які заважають електромагнітному полю.

					КС КРБ 123.023.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

— Схильні до великих похибок і тремтіння через часті вимоги до калібрування.

1.3 Вимоги до системи відтворення руху людини у віртуальному просторі

Захоплення рухів людини та передачі їх у віртуальний простір за допомогою VR трекерів побудованих на базі сенсорів гіроскопа та акселерометра, без використання додаткових станцій з камерами. Трекери мають коштувати не дорого, а саме у районі 3 000грн за шт. Така ціна є малою відносно аналогів від компанії Vive. Розробка плагіну для програми Steam VR у якому можна буде об'єднати усі трекери та надавати їм роль певних части тіла людини. Підключення до ПК має виконуватися за допомогою бездротової мережі Wi-Fi.

Захоплення рухів повинно бути як омога точним наскільки цього можна добитися без використання додаткових станцій з камерами а тажож, ефект дріфту трекера має буди мінімальним.

					КС КРБ 123.023.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

РОЗДІЛ 2 ПРОЄКТНА ЧАСТИНА

2.1 Структура системи захоплення рухів в VR

Принцип роботи наведеної структури (рис.2.1) полягає у тому що у VR трекер буде збирати дані з сенсора гіроскопа та акселерометра, зібрані дані будуть відправлятися по бездротові мережі Wi-Fi до ПК за допомогою вбудованого Wi-Fi модулю. Після цього передані дані надходять у плагін підключаючи таким чином трекер до ПК, та прив'язуючи трекер до Steam VR а Steam VR в свою чергу з'єднує трекари з VR шоломом.



Рисунок 2.1 – Структура системи захоплення рухів в VR

2.2 Розробка планіну для Steam VR

Програма Steam VR яка була розроблена компанією Valve є наразі єдиною програмою яка може забезпечити роботу VR шолому та VR трекерів від різних офіційних виробників таких як HTC VIVE які знову ж таки, через дуже малий вік VR індустрії є практично монополістами на ринку VR трекерів.

					КС КРБ 123.023.00.00 ПЗ					
Змн.	Арк.	№ докум.	Підпис	Дата	Проектна частина			Літ.	Арк..	Акрушів
Розроб.		Соколов О.М.						19	57	
Перевір.		Жаровський Р.О						ТНТУ, каф. КС, гр. СІзс-42		
Рецез.		Никитюк В. В								
Н. контр.		Тиш Є. В								
Затв.		Осухівська Г.М.								

Steam VR дуже зручна у використанні, вона сама здатна автоматично об'єднати усе в одну систему а це VR шолом, контролери для рук (рис. 2.2), VR трекери, та станцій з камерами. Та що є най важливішою з можливостей програми Steam VR це моддинг. Це дасть змогу модифікувати нашу копію Steam VR власним плагіном.



Рисунок 2.2 — Вікно програми Steam VR

На щастя станції з камерами нам не знадобляться. Оскільки трекери які будуть створені у цьому проєкті не є офіційними, а є свого роду DIY розробкою, програма Steam VR просто не буде їх розпізнавати, для цього нам і потрібен плагін. Плагін для Steam VR буде виконувати роль сервера який буде з'єднувати трекери з ПК, буде давати можливість їх налаштувань, надання ролі частини тіла, позиції, орієнтації та їх калібрування.

А також надасть змогу Steam VR розпізнавати наші трекери щоб у подальшому їх можна було використовувати у іграх або інших соціальних платформах для VR.

Нище буде наведено алгоритм роботи плагіну (рис.2.3).

					КС КРБ 123.023.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

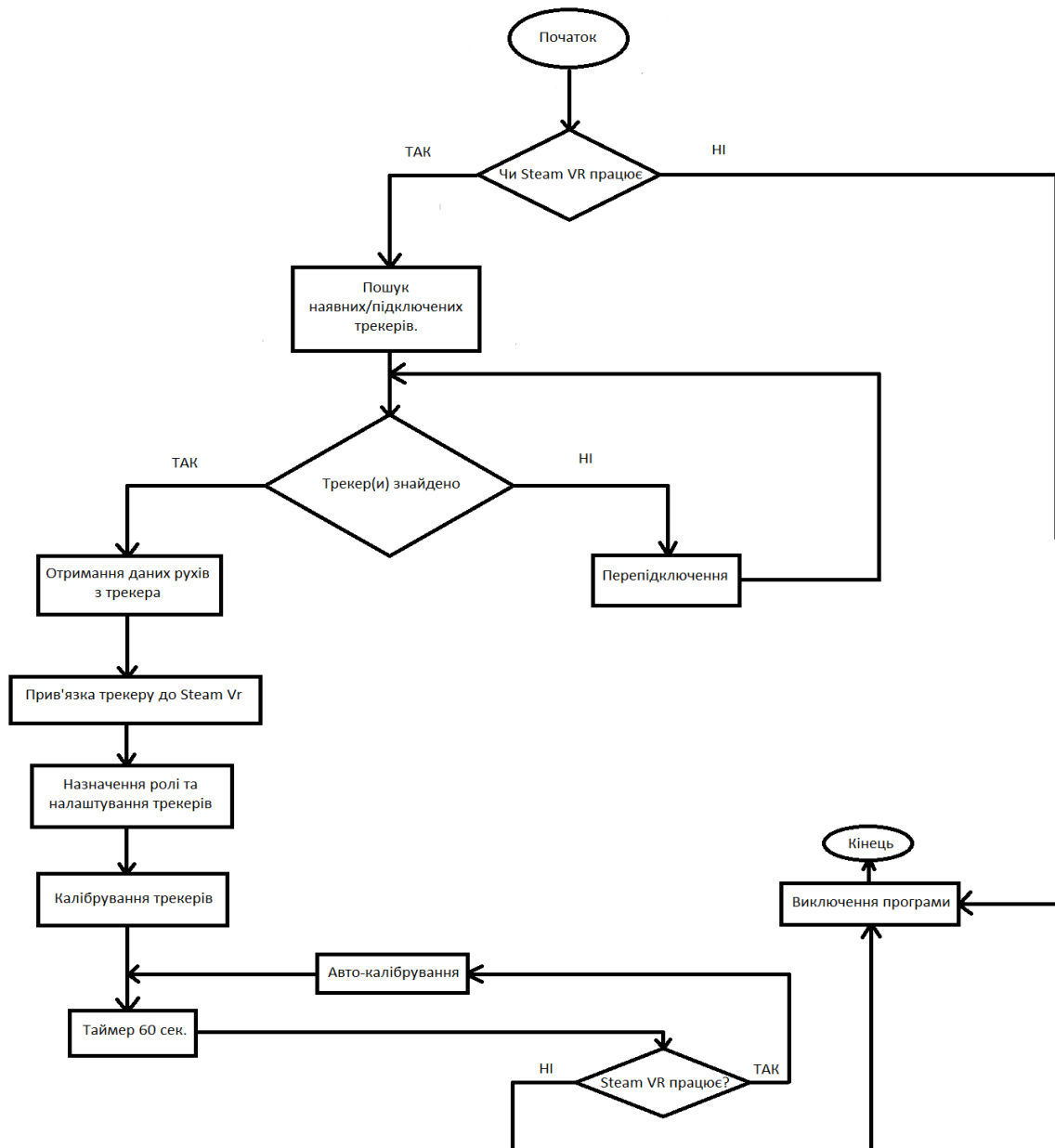


Рисунок 2.3 — Алгоритм роботи плагіну для Steam VR

2.3 Передача даних за допомогою Wi-Fi

Для передачі інформації бездротовим каналом було вибрано доволі хороший WI-FI модуль по відношенню ціна та якість WeMos D1 mini, доволі хороший подуль з просунутими технологіями, з модулем малих розмірів ESP-12E. Підключається просто до встроеного USB роз'єму що допомагає уникнути додавання лишніх USB що в свою чергу допомагає не збільшувати розміри трекера, також плата має вбудований чіп USB моста CH340G.

Сам модуль має дуже багато компіляторів, а також підтримується програмування в програмі Arduino IDE що буде дуже важливим у майбутньому. Цей WiFi модуль часто використовується в DIY проєктах, для створення інтернету речей, у робототехніці, віддаленого моніторингу та ін. А саме головне, модуль має не великі розміри всього 34 x 25 мм, що для трекера є одним з дуже важливих показників.

Плата чудово програмується в Arduino, та підтримує як послідовне, так і ОТА-програмування.

Порт CH340, в основі якого лежить USB-чіп, розташований на одній стороні міні-плати ESP8266, там також розташована кнопка перезавантаження. Роз'єми на платі розташовані так, щоб зручно встановити відразу кілька периферійних пристроїв уникаючи запутання кабелів. А середовище розробки Arduino, по якому зараз є багато інформації в мережі, допоможе легше налаштувати Wi-Fi модуль.

2.4 Калібрування трекерів у VR

Калібрування трекерів в VR для системи Full-body дещо відрізняється від звичного визначення калібрування інших вимірювальних приладів, у тому що калібрування у VR, означає під собою синхронізацію розташування трекерів на тілі людини з частинами тіла аватара в віртуальному просторі, так як ролі трекерам а саме чи то рука, нога і тд. буде назначено автоматично програмою Steam VR яка використовується для роботи практично усіх шоломів віртуальної реальності на даний період часу. Під час калібрування слід упевнитися що похибка розташування трекера у віртуальному просторі була як найбільш меншою, щоб досягнути максимально більш натурального передання рухів людини на аватар.

VR-трекери, зазвичай проходять процес калібрування, щоб забезпечити точне відстеження у віртуальному середовищі.

					КС КРБ 123.023.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

Ось більш детальніше пояснення як калібруються VR трекери:

1. Налаштування та позиціонування: Щоб почати процес калібрування, потрібно налаштувати систему віртуальної реальності і переконатися, що базові станції (Якщо вони є в наявності) правильно встановлені і налаштовані. Базові станції малюють невидиму для людського ока сіточку або дуже велику кількість точок, які приймаються датчиками на VR-трекерах.

2. Оновлення прошивки та програмного забезпечення: Важливо постійно оновлювати прошивку і програмне забезпечення як VR-трекерів, так і системи віртуальної реальності. Це гарантує, найновіші функції та вдосконалення, включаючи будь-які покращення калібрування.

3. Програмне забезпечення для калібрування: Системи віртуальної реальності, такі як HTC Vive, часто надають спеціальні програмні інструменти для допомоги в процесі калібрування. Ці програмні інструменти допоможуть оптимізувати роботу трекера.

4. Вирівнювання трекера: Під час калібрування зазвичай з'являється запит на утримання або розміщення трекерів віртуальної реальності в певних положеннях і орієнтаціях. Це дозволяє системі встановити еталонне початкове положення та орієнтацію трекера у віртуальному середовищі.

5. Діапазон і покриття: Калібрування може передбачати переміщення VR-трекерів у межах ігрової зони, щоб переконатися, що вони перебувають —

— у зоні дії та покриття базових станцій. Це допомагає встановити сильний і стабільний сигнал відстеження по всьому об'єму відстеження. Перевірка та точне налаштування: Після початкового калібрування система може надавати кроки перевірки для забезпечення точного відстеження. Це може включати перевірку вирівнювання і руху трекерів щодо віртуальних об'єктів або візуальних маркерів у середовищі віртуальної реальності. Регулярні перевірки: Рекомендується періодично перевіряти і повторно калібрувати VR-трекери (нажаль в ручну), особливо якщо буде помічено будь-які —

— невідповідності у відстеженні або зміни в продуктивності відстеження.

					КС КРБ 123.023.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

Це допомагає підтримувати точні та надійні результати відстеження з часом.[7]

2.5 Апаратна реалізація

2.5.1. Варіанти схем для VR трекера

Перший варіант схеми трекера VR зроблена на базі сенсора руху з гіроскопом та акселерометром MPU-6050 (рис.2.4).

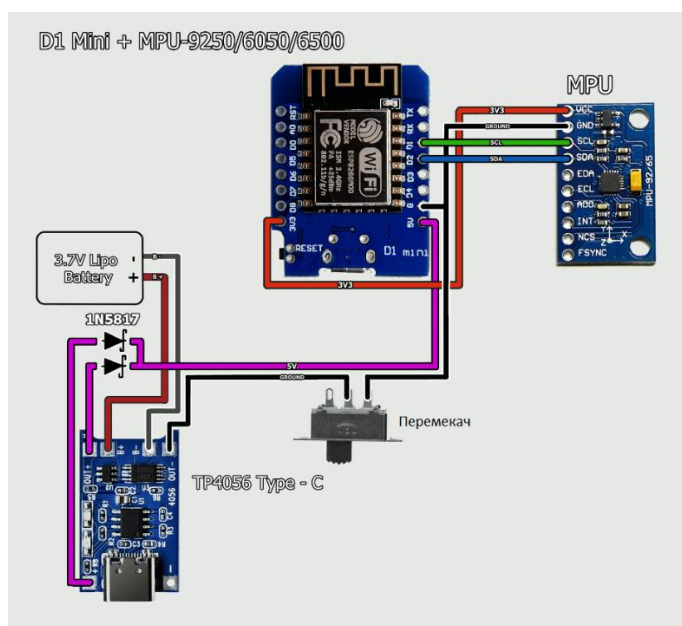


Рисунок 2.4 — Схема VR трекера на базі MPU-6050

Сенсор MPU-6050 є най дешевшим варіантом з усіх, в інтернет магазинах його можна знайти всього лиш за 115 грн. Хороший сенсор за свою ціну, але його відстеження не є дуже точними, також є не приємний ефект дріфту, тобто сенсор по трішки зсувається в сторону через що преїдеться доволі часто перекалібрувати аватар в віртуальному просторі. Нище наведено його характеристики:

- Живлення 3.3 - 5.0 В.
- Чіп MPU-6050.

					КС КРБ 123.023.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

- Діапазон гіроскопу $\pm 250 \pm 500 \pm 1000 \pm 2000$ $\%$ / с.
- Діапазон акселерометру $\pm 2 \pm 4 \pm 8 \pm 16$ g.
- Протокол ІІС (I2C / TWI).
- Вбудований 16-бітний АЦП.
- Ромір плати 20 x 16 мм.

Другий варіант схеми трекера на базі сенсору BMI160 (рис. 2.5).

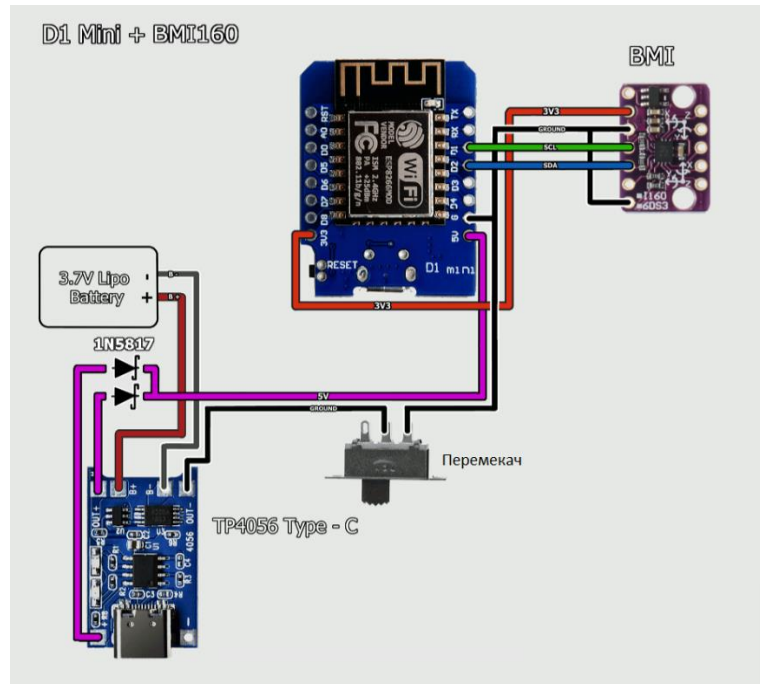


Рисунок 2.5 — Схема VR трекера на базі BMI160

Маленький сенсор BMI160 є хорошим модулем по відношенню ціна/якість, в інтернеті цей сенсор можна знайти за ціною 240 грн. Має доволі низкий рівень споживання струму що може надати довший час роботи трекера який буде живитися від невеликого акумулятора а також має вбудований 16-розрядний IMU з досить низьким рівнем шуму. Дріфт сенсора доволі малий а відстеження сенсора у просторі задовільно точне. Досить не поганий та бюджетний сенсор для VR трекера.

Характеристики:

- Живлення 1.2 - 3.6 В.

					КС КРБ 123.023.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

- Чіп BMI160.
- Діапазон гіроскопу ± 125 °/с: 262.4 LSB/°/с ± 250 °/с: 131.2 LSB/°/с ± 500 °/с: 65.6 LSB/°/с ± 1000 °/с: 32.8 LSB/°/с ± 2000 °/с: 16.4 LSB/°/с.
- Діапазон акселерометру $\pm 2g$: 16384 LSB/g $\pm 4g$: 8192 LSB/g $\pm 8g$: 4096 LSB/g $\pm 16g$: 2048 LSB/g.
- Протокол I2C.
- Температура -40 +85 °С.
- Ромір плати 2.5x3.0x0.8мм³.

Третій варіант схеми трекера на базі BNO085 (рис. 2.6).

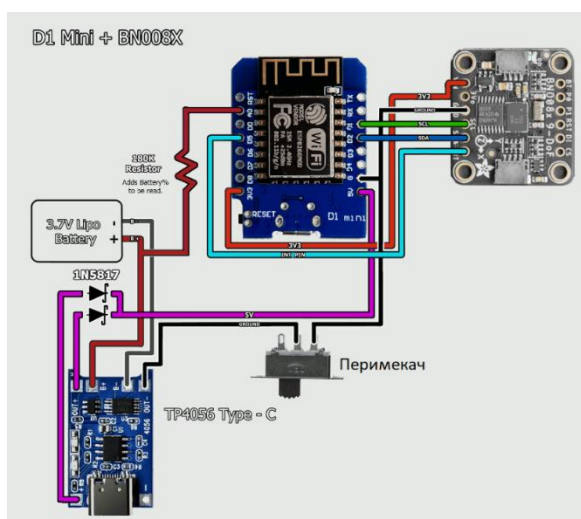


Рисунок 2.6 — Схема VR трекера на базі BNO085

Сенсор BNO085 (рис. 2.7) є дорогим, але ідеальним варіантом для VR трекера, його ціна в інтернет магазинах становить 2 250 грн. У цього сенсора практично відсутній ефект дріфту сенсора що є просто чудовим показником для VR трекера, а відсліджування у просторі дуже точне. Трекер на базі такого сенсора вже може становити якусь конкуренцію трекерам Vive які коштують 6 300 грн, плюс використовують станції з камерами і ні, станції не входять в ціну трекерів Vive. Тому використовуючи BNO085 можна зробити відносно дешевий трекер який то того ж буде працювати автономно без допомоги станцій з камерами.

					КС КРБ 123.023.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

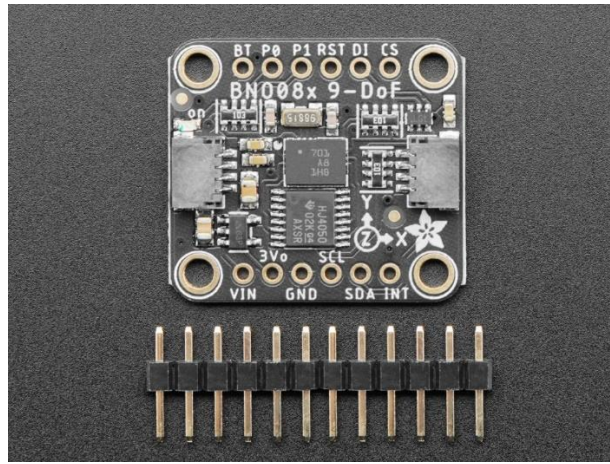


Рисунок 2.7 — Сенсор руху BNO085

Особливості даного сенсора:

- Вектор прискорення / Акселерометр.
- Три осі прискорення (гравітація + лінійний рух) в m/s^2 .
- Вектор кутової швидкості / Гіроскоп.
- Три осі "швидкості обертання" в рад/с.
- Вектор напруженості магнітного поля / магнітометр.
- Три осі вимірювання магнітного поля в мікротеслах (мкТл).
- Вектор лінійного прискорення.
- Три осі даних лінійного прискорення (прискорення мінус сила тяжіння) в m/s^2 .
- Гравітаційний вектор.
- Три осі гравітаційного прискорення (мінус будь-який рух) у m/s^2 .
- Вектор абсолютної орієнтації / обертання.
- Чотириточковий кватерніонний вивід для точної маніпуляції даними.

BNO085 також надає вражаючий набір продуктів для виявлення та класифікації, додатково аналізуючи виміряний рух:

- Виявлення та класифікація стабільності.
- Детектор натискання.

					КС КРБ 123.023.00.00 ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

- Детектор кроків.
- Лічильник кроків.
- Класифікація активності.
- Детектор значного руху.
- Датчик струсу.

Піни живлення:

— VIN - це контакт живлення. Оскільки мікросхема датчика використовує 3 В постійного струму, ми включили на плату стабілізатор напруги, який візьме 3-5 В постійного струму і безпечно перетворить його в постійний. Для живлення плати подайте на неї таку ж напругу, яка відповідає логічному рівню вашого мікроконтролера - наприклад, для 5В мікроконтролера, такого як Arduino, використовуйте 5В

— 3Vo - це вихід 3,3 В з стабілізатора напруги, ви можете взяти з нього до 100 мА, якщо хочете

— GND - загальна земля для живлення та логіки

— Логічні виводи I2C

— SCL - вивід тактової частоти I2C, підключається до тактової лінії I2C вашого мікроконтролера. Цей вивід зміщений за рівнем, тому ви можете використовувати логіку 3-5В, і на цьому виводі є підтяжка 10К.

— SDA - вивід даних I2C, підключається до лінії даних I2C вашого мікроконтролера. Цей вивід зміщений за рівнем, тому ви можете використовувати логіку 3-5В, і на ньому є підтяжка 10К.

— STEMMA QT - безпаяльна альтернатива контактам Vin, GND, SCL і SDA. Ці роз'єми дозволяють підключати до розробницьких плат роз'єми STEMMA QT і різні супутні аксесуари.

— DI - вивід адреси I2C. Замикання цього виводу на високий рівень змінює адресу I2C з 0x4A на 0x4B.

Схема сенсора BNO085 показана на (рис 2.8).

					КС КРБ 123.023.00.00 ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

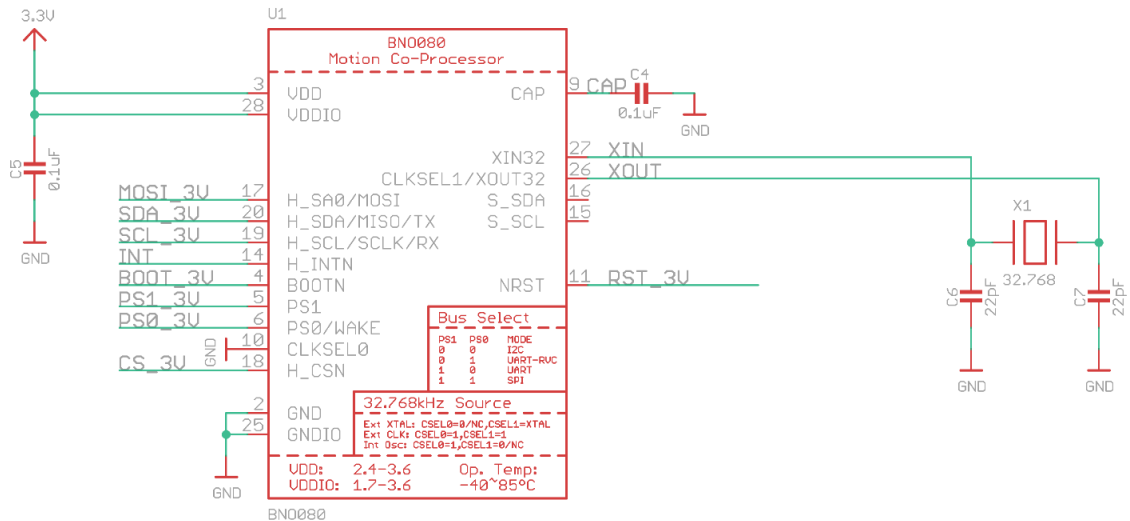


Рисунок 2.8 — Схема сенсор руху BNO085

Логічні виводи UART:

— SCL - вхід даних UART до датчика - підключіть до виводу TX вашого мікроконтролера.

— SDA - вихід даних UART з датчика - підключіть до виводу RX вашого мікроконтролера.

Логічні виводи SPI:

— SCL - це також вивід SPI Clock / SCK, це вхід для мікросхеми.

— SDA - Подвійний як вихід даних / вхід мікроконтролера / вихід датчика / вивід MISO, для даних, що надсилаються з BNO08x.

— DI - вхід даних / вихід мікроконтролера / вхід датчика мікроконтролера / вивід MOSI, для даних, що надсилаються з вашого процесора на BNO08x.

— CS - це вивід Chip Select, потягніть його вниз, щоб почати SPI-транзакцію. Це вхід для мікросхеми.

— INT - Переривання - активний низький рівень. Вказує на те, що BNO085 потребує уваги хоста. Необхідний для стабільної роботи SPI.

— RST - Скидання - активний низький рівень - подайте низький рівень на GND, щоб скинути датчик. Необхідний для стабільної роботи SPI.

Далі розписані інші компоненти схеми VR трекера.

2.5.2. Зарядний модуль TP4056 Type-C

Модуль контролера із захистом TP4056 (рис. 2.9, рис. 2.10) призначений щоб заряджати літійно іоні акумулятори струмом не вище 1А. Мала кількість деталей на платі роблять TP4056 дуже хорошим вибором для трекерів VR так як їх потрібно зробити як омога меншими. Особливістю цього модуля є присутність на платі USB TYPE-C який широко використовується усюди, особливо у VR індустрії.

Для зарядки TP4056 потрібно підключити USB-кабель у підготовлений для цього роз'єм. Також слід приєднати 5В на контакти (IN+) (IN-), приєднавши акумулятор (розряджений) до контактів (B+) (B-), навантаження, по якому буде заряджатися акумулятор, слід приєднати до контактів OUT+ OUT-. Коли зарядний модуль з акумулятоором дуже приєднано до мережі живлення для підзарядки акумулятара слід впевнитися що на модуль зарядки не буде ніякого іншого навантаження. Зелена лампочка на платі буде вказувати на те що зарядка акумулятора завершена, а червона лампочка буде вказувати що акумулятор розряджений, або в процесі зарядки.

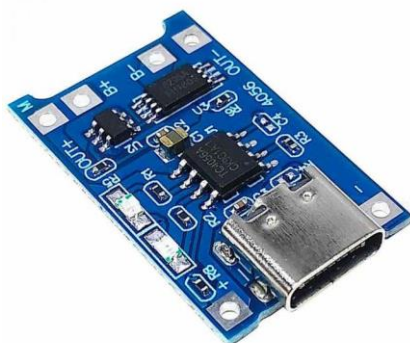


Рисунок 2.9 — Зарядний модуль TP4056 Type-C

					КС КРБ 123.023.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		30

Мінусом створеної зарядної системи може бути те що під час заарядки плата може нагріватися до 100 °С. Це не є якоюсь поломкою, просто така невелика особливість та неприємність зарядної плати.

Характеристики:

- Вхідна напруга: 4.5 – 5.5В.
- Кінцева напруга зарядки: 4.2В.
- Напруга захисту Розряду: 2.4В.
- Ток зарядки: 1А.
- Роз'єм підключення ЗУ : Type-C.
- Діапазон робочих температур :-10° до + 85°.
- Розмір: 27,75x17,25 мм.

Щоб налаштувати силу струму під час зарядки потрібно замінити резистор R3 (1,2кОм) на 3 контакті TP4056, по стандарту: 1А

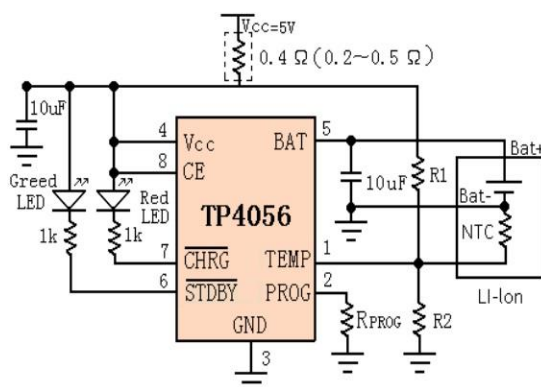


Рисунок 2.10 — Зарядний модуль TP4056 Type-C

Особливості:

- Програмований струм заряду до 1000 мА.
- Без MOSFET, сенсорного резистора або блокування.
- Потрібен діод.
- Повний лінійний зарядний пристрій в SOP-8.
- Акумулятори:
- Постійний струм / постійна напруга.

					КС КРБ 123.023.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

— Заряджає одноклітинні літій-іонні акумулятори безпосередньо від порту USB.

— Попередньо встановлена напруга заряду 4,2 В з похибкою 1,5%.

Точність:

— Автоматична перезарядка.

— Два вихідних виводу стану заряду.

— Завершення заряду -С/10.

— Поріг протікання заряду 2,9 В (TP4056).

— Плавний пуск обмежує пусковий струм.

— Доступний радіатор у 8-вивідному корпусі SOP, радіатор потрібн підключити до заземлення або насувається.

2.5.3. Wi-Fi модуль ESP8266 WeMos D1 Mini

Плата WeMos D1 mini (рис. 2.11) є аналогом плат Witty Cloud та NodeMCU v3, що найважливіше для віар трекерів плата має дуже малі розміри, що є однією з найголовніших причин вибору саме цієї плати а також на платі встановлений модуль ESP8266 версії ESP-12F. WeMos D1 mini не потребує зовнішнього мікроконтролера, через вбудований 32-бітний мікроконтролер тактова частота якого 80 МГц та чіп з пам'ятю в розмірі 4 МБ.

Модуль WeMos D1 mini – дуже гарно підходить для систем віддаленого управління а саме головне автономних датчиках які і будуть головною складовою трекерів.

Працювати з Wi-Fi модулем ESP8266 WeMos D1 mini доволі комфортно, тому що в модулі є вбудований USB-UART конвертер CH340, та стабілізатор напруги, контакти виведені на роз'єм зі стандартним кроком 2,54 мм.

					КС КРБ 123.023.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32



Рисунок 2.11 — Wi-Fi модуль ESP8266 WeMos D1 Mini

Характеристики:

- Підтримувані мережі 802.11 b/g/n.
- Частота мікроконтролера 80 МГц.
- Швидкість UART 115200.
- Робоча напруга 3.3 В.
- Напруга живлення 3.7-12 В.
- Максимальний струм споживання (під час передавання) 240 мА.
- Розміри 33 x 26x 6 мм.

ESP8266 WeMos D1 Mini має лише один вихідний вивід 3,3 В і лише один вивід заземлення для живлення зовнішніх ланцюгів. Сьогодні легко створювати кастомні друковані плати, які не є дорогими. Під час створення проекту потрібно з'єднати 3,3 В і заземлення з рейками макетної плати для живлення різних електронних компонентів.

Вивід 5В може бути використаний як джерело живлення для ESP8266 з напругою від 4 до 6 В або як вихід 4,67 В, коли ESP8266 WeMos D1 Mini живиться через USB. Таким чином, можна використовувати електронні компоненти, які потребують напруги живлення 5 В, оскільки ці компоненти також повинні працювати з напругою живлення 4,67 В.

					КС КРБ 123.023.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

ESP8266 WeMos D1 Mini має в цілому 9 цифрових входів / виходів (рис. 2.12). Всі ці 9 виводів можуть створювати ШІМ-сигнали і мають максимальний вихідний струм 12 мА. На жаль, ESP8266 має лише 1 аналоговий вхід А0, який з'єднаний з 10-бітним аналого-цифровим перетворювачем (АЦП) для перетворення аналогової напруги в 1024 цифрових значення між 0 і 1023.

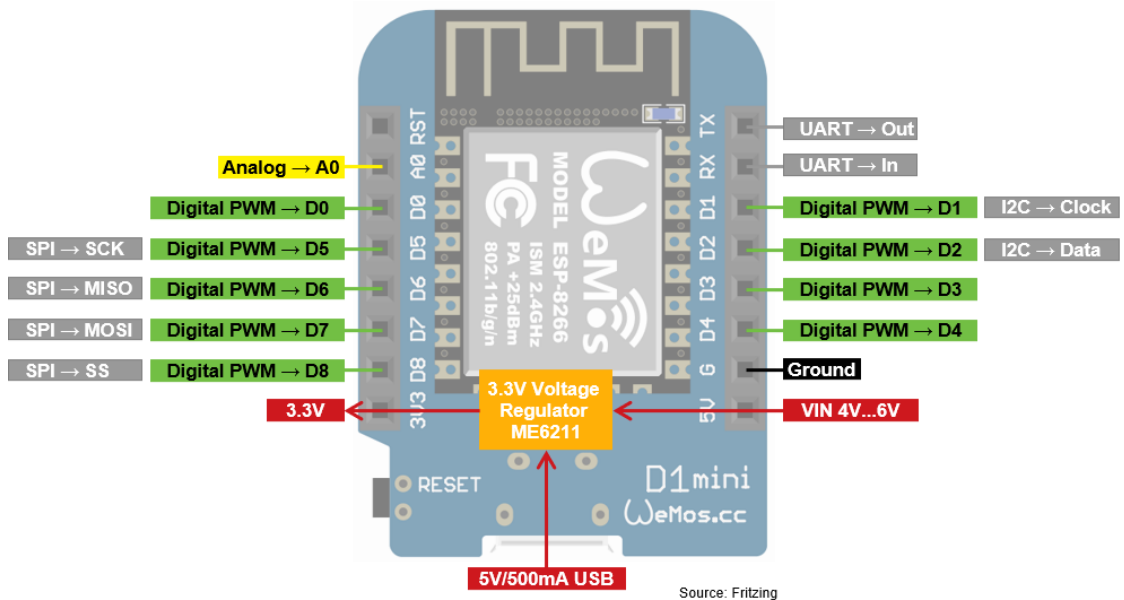


Рисунок 2.12 — Піни ESP8266 WeMos D1 Mini

Міні-блок живлення ESP8266 WeMos D1.

Блок живлення ESP8266 WeMos D1 Mini залежить від різних рівнів напруги мікроконтролера. Тому ми повинні знати, які електронні компоненти відповідають за різні рівні напруги. На наступному малюнку представлений огляд рівнів напруги і максимальних струмів WeMos D1 Mini.

					КС КРБ 123.023.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

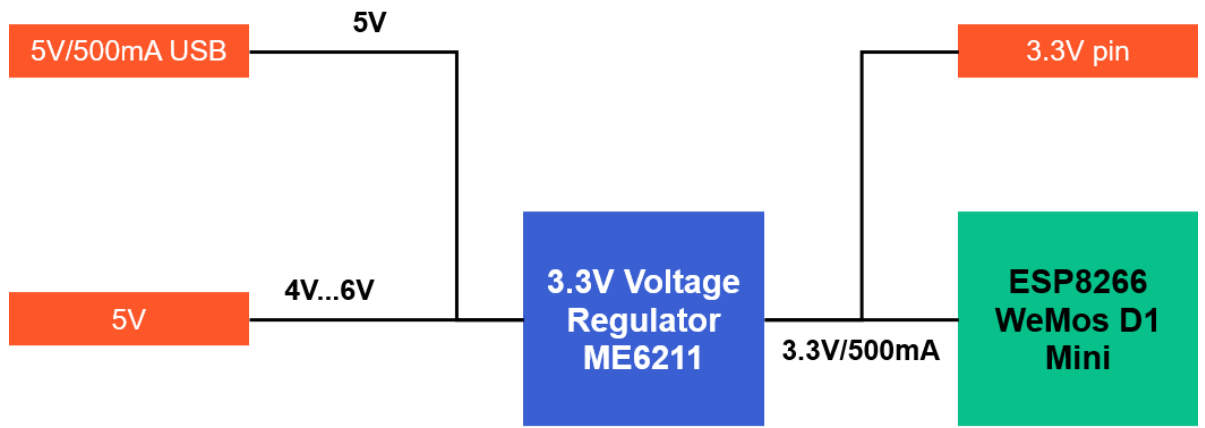


Рисунок 2.13 — Схема живлення ESP8266

Як жити ESP8266 WeMos D1 Mini.

Буде використано 5V Pin. Тому що трекер матиме зовнішнє джерело живлення, батарейки або акумулятори. Напруга повинна бути між 4,3В і 6В. Таким чином, можна жити WeMos D1 Mini від акумулятора з роз'ємом JST в поєднанні з екраном для батареї для WeMos D1 Mini.

Батарей

Існує багато різних варіантів акумуляторів, і розмір. Буде використано один акумулятор для Wemos D1 Mini. Що стосується часу автономної роботи, то кожні 100 мАг, на які розрахована батарея, вистачає на 1 годину використання. Отже, акумулятора ємністю зазвичай вистачає приблизно на 12 годин безперервної роботи.

Плоскі Li-Po акумулятори, як правило, досить правдиво вказують свою ємність, проте ємність 18650 може сильно відрізнятись в залежності від виробника. Типовий безіменний 18650 може мати низьку ємність 800 мАг, тоді як 18650, виготовлений відомим виробником, таким як LG, Samsung або Sony, може мати високу ємність 3500 мАг. Загалом, слід з підозрою ставитися до заявленої ємності 18650.

Варто також зазначити, що не всі акумулятори однакові. Деякі з них мають схему захисту, а деякі ні. Взагалі кажучи, плоскі Li-Po акумулятори зазвичай мають схему захисту, в той час як акумулятори типу 18650 її не мають.

					КС КРБ 123.023.00.00 ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

Li-Po акумулятор, схема захисту зазвичай виглядає як невелика друкована плата з кількома мікросхемами під жовтою поліімідною стрічкою на акумуляторі. З огляду на це, зарядна плата TP4056 забезпечить ті ж самі функції, тому відсутність захисту акумулятора не повинна бути фактором, що може призвести до поломки, а скоріше додатковим душевним спокоєм і додатковою безпекою.

Рекомендації щодо прокладання кабелю для допоміжного трекера.

Звернемо увагу, що хоча на схемі показано, що SDA і SCL проходять поруч, слід переконатися, що вони фізично не знаходяться поруч, коли кабель допоміжного трекера буде прокладений. (рис.2.14).

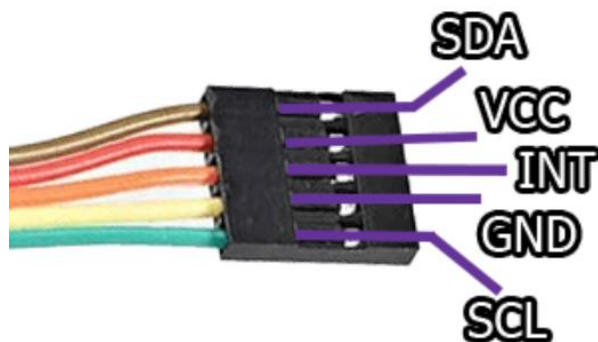


Рисунок 2.14 — Розташування дротів на стрічковому кабелі

Це необхідно для уникнення перехресних перешкод і забезпечення стабільної роботи обох трекерів, коли кабель дозволяє безпечно подовжувати його на відстань понад 80 см. Було використано стрічковий кабель, росташуємо дроти так як на (рис.2.14).

РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА

3.1. Програмна реалізація

3.1.1 Встановлення бібліотеки та прошивка BNO085(сенсор)

Встановлення бібліотеки Adafruit BNO08x було виконано за допомогою менеджера бібліотек в Arduino IDE. Для цього ми відкриваємо менеджер бібліотек, клікаємо на Sketch—Include Library---Manage Libraries (рис.3.1).

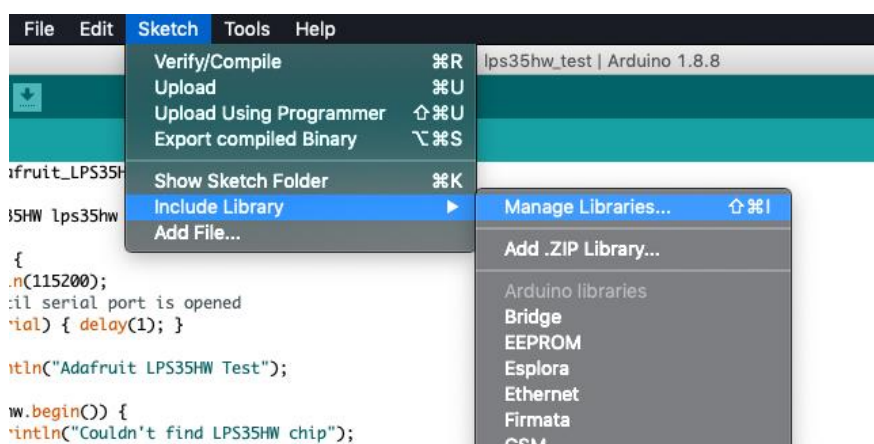


Рисунок 3.1 — Пошук бібліотеки

Натискаємо пункт меню Manage Libraries..., шукаємо Adafruit BNO08x і виберіть бібліотеку Adafruit BNO08x (рис. 3.2).

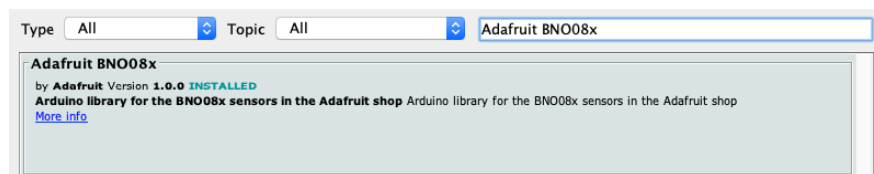


Рисунок 3.2 — Вибір бібліотеки Adafruit BNO08x

Змн.	Арк.	№ докум.	Підпис	Дата	КС КРБ 123.023.00.00 ПЗ			
Розроб.		Соколов О.М.			Практична частина	Літ.	Арк..	Акрушів
Перевір.		Жаровський Р.О					37	57
Рецез.		Никитюк В. В				ТНТУ, каф. КС, гр. СІзс-		
Н. контр.		Тиш С. В				42		
Затв.		Осухівська Г.М.						

Після того як описані вище дії було виконано у тому ж вікні пишемо код який дасть змогу зчитувати дані отримані сенсорами(гіроскопа та акселерометра) а саме, позицію в просторі і повороти осі (див. Додаток Б 1).

3.1.2 Прошивка коду на ESP8266 WeMos D1 Mini

Для прошивки Wi-Fi модуля знову було використано Arduino IDE і використовувати Arduino IDE. Існує лише невелика зміна в Arduino IDE, яка описана в наступних кроках:

1. У меню Arduino IDE натисється Файл → Налаштування Вставте наступну URL-адресу в поле "Additional Boards Manager URLs:": http://arduino.esp8266.com/stable/package_esp8266com_index.json.

На наступному малюнку(рис.3.3) показано результат у меню налаштувань.

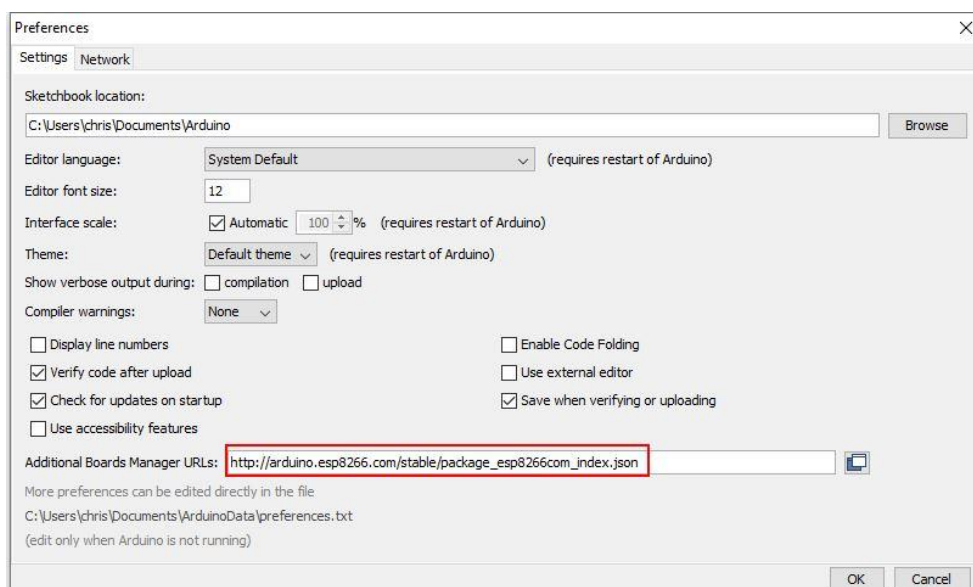


Рисунок 3.3 — Введення посилання на встановлення бібліотеки

2. Встановлення необхідних пакетів для модуля ESP8266:

Потрібно натиснути в Arduino IDE на Tools → Board → Board Manager. Після чого відкривається NodeMCU або ESP8266 і можна знайти esp8266 від ESP8266 Community. Встановлюється остання версія плати. (Див рис 3.3 та 3.4).

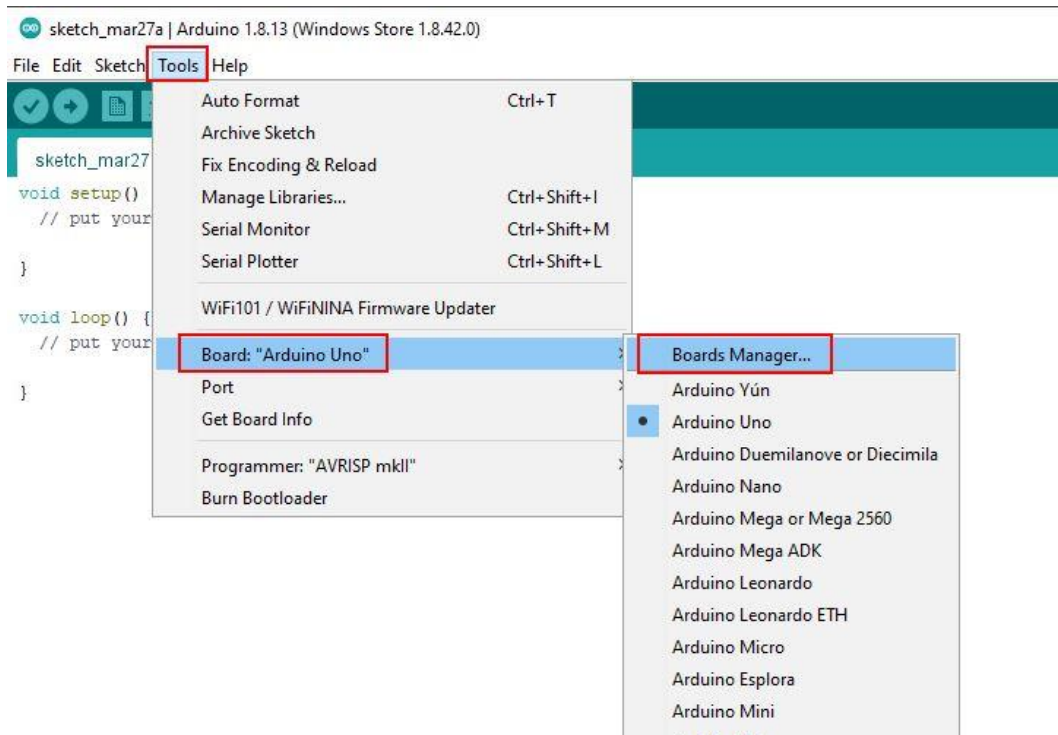


Рисунок 3.4 — Шлях до вибору прошивки плати Wi-Fi

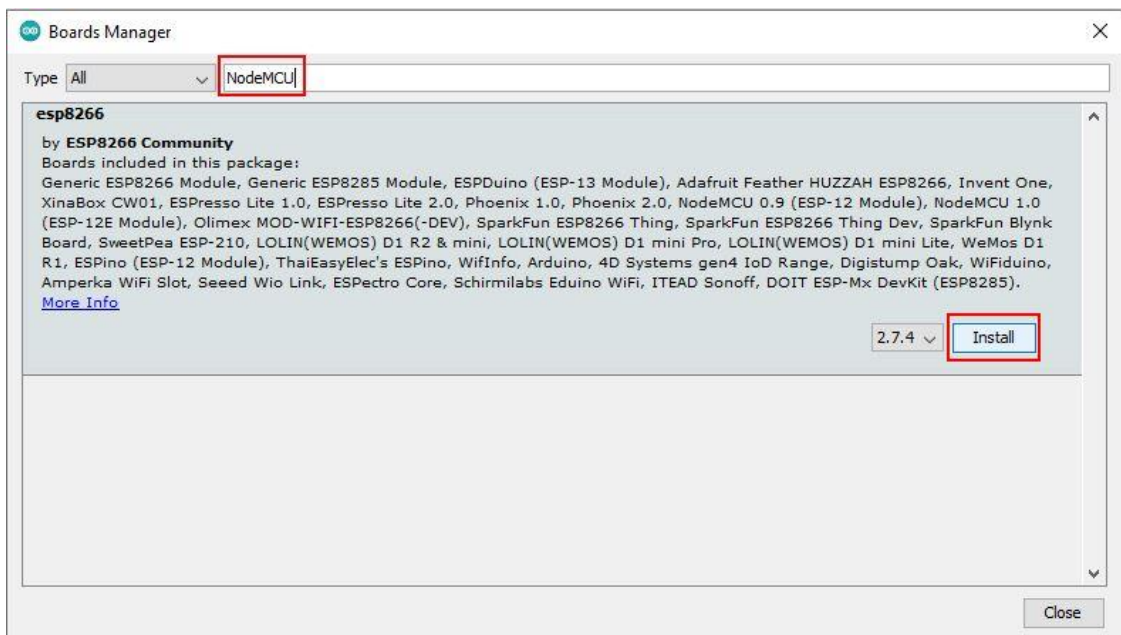


Рисунок 3.5 — Вибір останньої прошивки плати Wi-Fi

Далі потрібно підключити плату до Wi-Fi роутера, для цього було написано наступний код у програмі Arduino IDE (див. Додаток Б 2).

3.2 Об'єднання та підключення трекера(ів) в одну систему (Full-body).

Для початку слід запустити програму Steam VR (рис 3.6) та упевнитися що віар шолом підключено до ПК, якщо шолом у справному стані він буде позначатися у вікні програми як зелений силует шолому приклад див. . Тому що без наявності віар шолому нічого у подальшому працювати не буде.



Рисунок 3.6 — Вікно програми Steam VR

Для того щоб підключити розроблений трекер до ПК було розроблено плагін який буде виконувати роль серверу куди трекери будуть підключатися а також буде об'єднувати трекери в одну систему якщо є більше а ніж один трекер, також передчається можливість надання конкретної ролі для трекера а саме за яку частину тіла від буде відповідати, чи то нога чи то талія і тд.

Запускаємо програму. Нас зустрінуть два вікна плагіну з великою кількістю інформації (рис.3.7) та консоль яка є сервером по якому буде передаватися уся інформація яку отримує трекер. Далі коротка інформація але важлива інформація що саме ми можемо побачити у вікні плагіну для Steam VR.

Під назвою «Tracker list» показана уся інформація про підключені та наявні трекери, а також саме там виконується надання ролі VR трекеру. З правої сторони вікна плагіну виконується ручне калібрування трекерів у просторі. Кнопка Reset у верху вікна плагіну виконує авто калібрування через 3 секунди після її натискання.

					КС КРБ 123.023.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

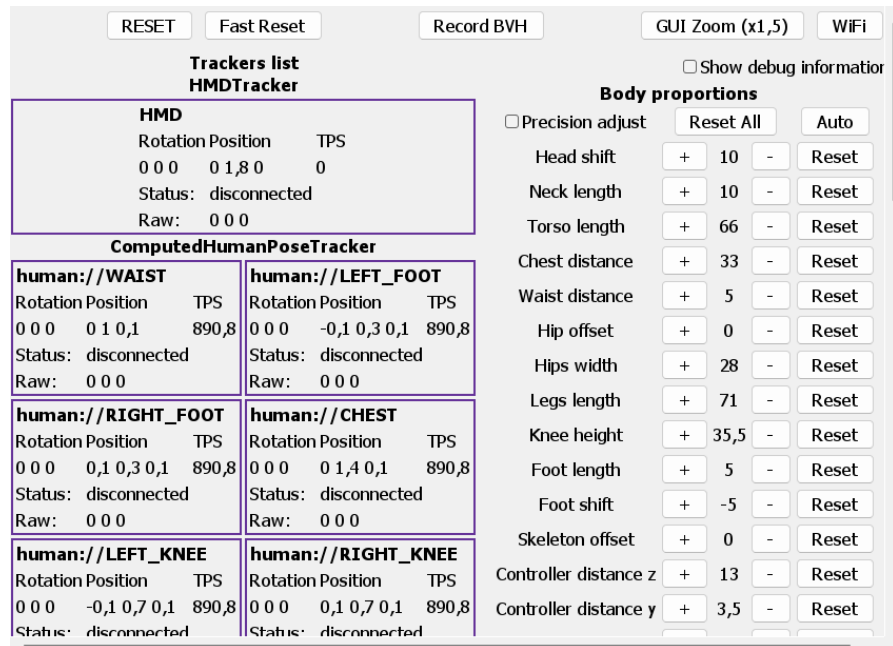


Рисунок 3.7 — Вікно плагіна для Steam VR

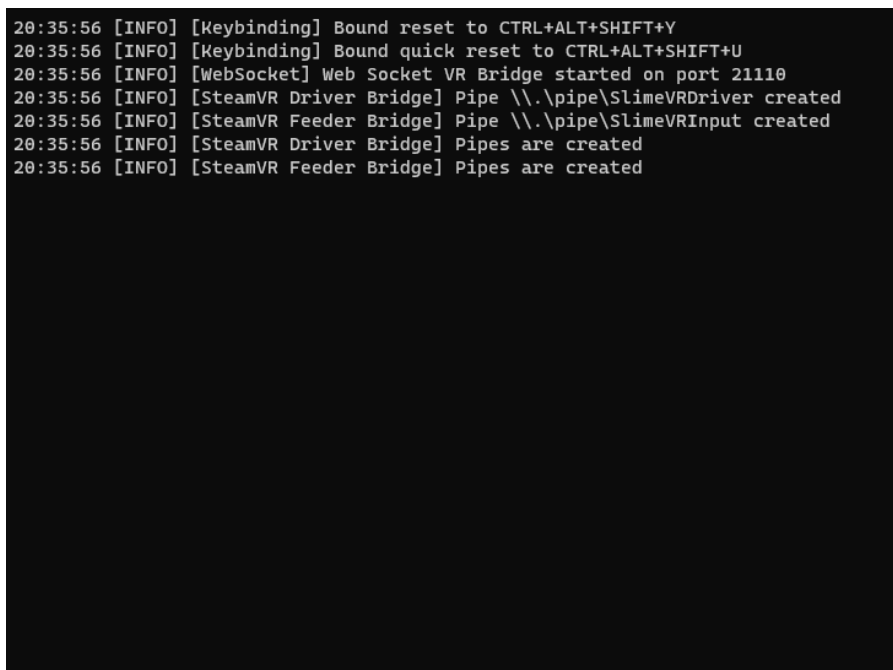


Рисунок 3.8 — Консоль сервера плагіна

Включаємо трекер. Як тільки трекер підключиться до без дротової мережі Wi-Fi, трекер відразу почне відображатися у програмі та в консолі.

Щоб упевнитися у тому що трекер підключився, в консолі має з'явитися наступна інформація див. рис. 3.9.

```

21:41:35 [INFO] [SteamVR Feeder Bridge] Pipes are created
21:41:35 [INFO] [SteamVR Driver Bridge] Pipes are created
21:41:43 [INFO] [TrackerServer] Handshake received from /192.168.1.1
09:9185
21:41:43 [INFO] [TrackerServer] Tracker 0 added with address /192.16
8.1.109:9185. Board type: 0, imu type: 0, firmware: owoTrack8 (8), m
ac: 43:1F:86:5A:67:96, name: udp://43:1F:86:5A:67:96
21:41:43 [INFO] [TrackerServer] Sensor 0 for udp://43:1F:86:5A:67:96
status: 1
21:41:43 [INFO] [TrackerServer] Added sensor 0 for udp://43:1F:86:5A
:67:96, type 0

```

Рисунок 3.9 — Трекер підключено до сервера

Далі потрібно настроїти трекер, а точніше задати йому роль, яку саме частину тіла він повинен відтворювати, це потрібно для того що коли буде запущено 3д віртуальний простір, програма розуміла на яку частину тіла віртуального аварата має передаватися інформація про рухи. У вінці програми нам потрібно можна буде побачити список активних трекерів. Тепер потрібно буде задати їм роль, приклад показано на (рис 3.10) задається роль талії/waist.

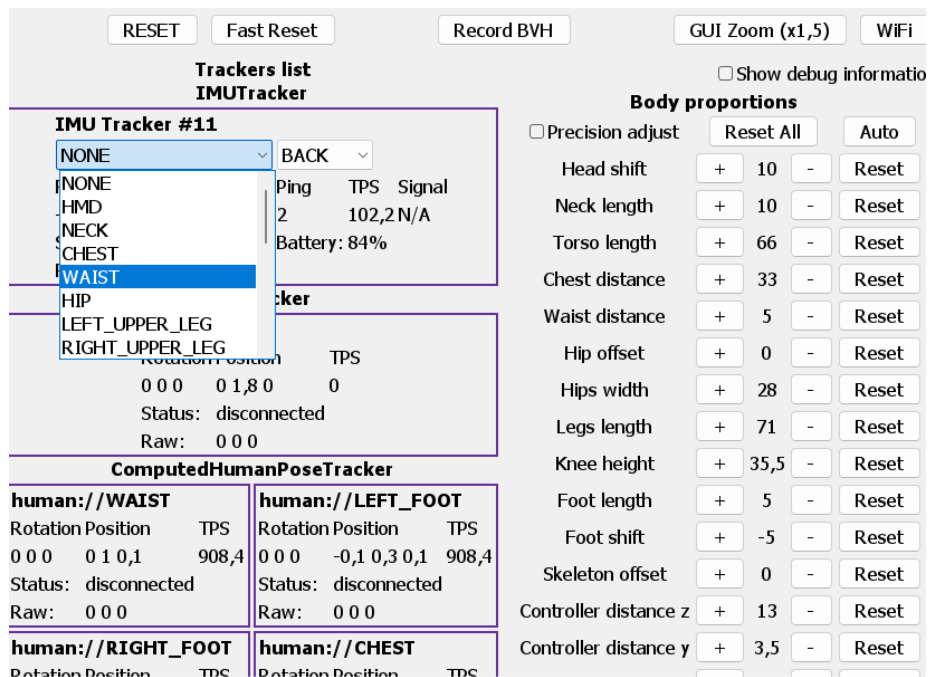


Рисунок 3.10 — Надання ролі для трекера

Таку процедуру потрібно буде зробити з кожним трекером на кожну частину тіла. Після того як це буде зроблено нам потрібно впевнитися що трекери працюють.

Щоб це перевірити, просто беремо один з трекерів у руку на крутимо і рухаємо його у просторі. У програмі на кожному трекері будуть мінятися числа під Rotation/Обертання та Position/Позиція (рис. 3.11).

```

human://WAIST
Rotation Position      TPS
-88 168 2 0,1 1,7 -0,5  912,8
Status: ok
Raw:      -88 168 2
    
```

Рисунок 3.11 — Дані про положення VR трекера

3.3 Перевірка працездатності зробленої системи Full Body у VR просторі.

Віар простір буде на рушії Unity. Для цього створюється новий проект вибираємо 3D основу після чого нажимаємо на кнопку «Create project» (рис. 3.12.).

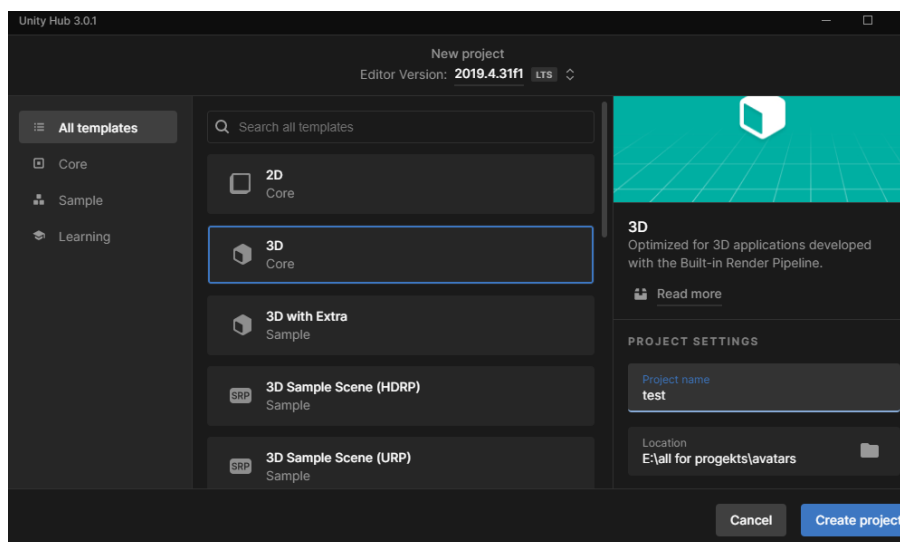


Рисунок 3.12 — створення проекту на рушії Unity

Далі імпортуємо в проект VRCS SDK3 це плагін за допомогою якого можна буде кикористовувати VR шолом та прив'язати трекери до аватару в проекті рушії Unity.

Для цього нам потрібно просто drag and drop файл з плагіном в панель з файлами та асетами проекту після чого нажати Імпортувати/Import, наглядно показано на (рис.3.13).

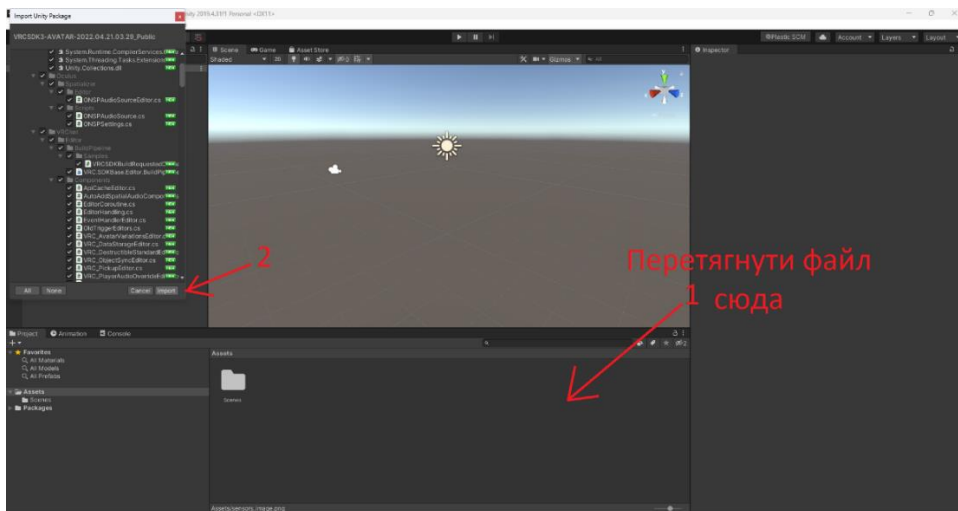


Рисунок 3.13 — Робоче середовище рушія Unity

Після того як плагін встановився потрібно створити пустий об'єкт. Кліктуни на нього лівою кнопкою, з правої сторони вікна є вкладка inspector, там потрібно натиснути на add component після чого знайти vrc avatar descriptor рис. (3.14).

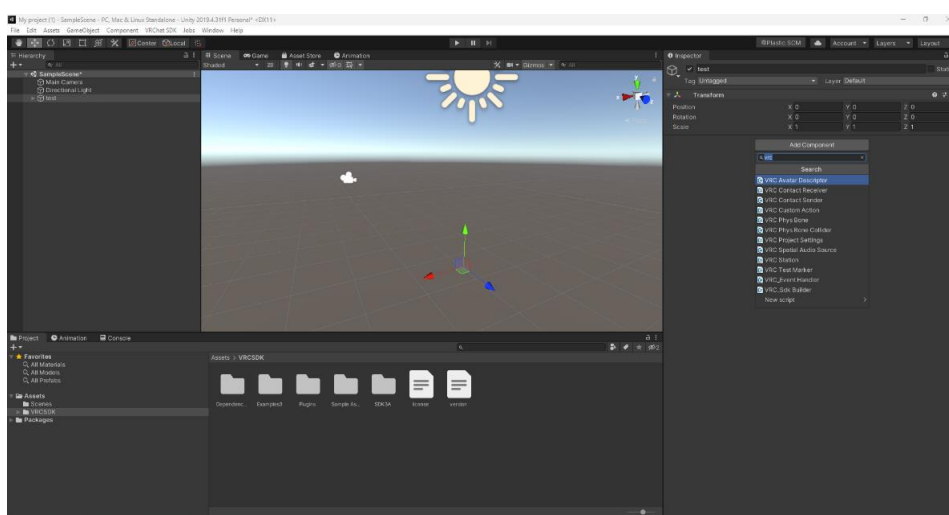


Рисунок 3.14 — Вибір компоненту

					КС КРБ 123.023.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

Після того як `vr avatar descriptor` було добавлено на об'єкт потрібно впевнитись що трекери усі працюють.

Для цього у потрібно натиснути на кнопку `select-configure`. Відкриється вікно де буде показаний чоловічок, якщо його усі частини тіла сelenого коляру значить усе працює, якщо якась частина тіла червоного коляру це означає що з трекером є якась проблема, та якщо якась частина тіла сірого коляру це означає що трекер для даної частини тіла повністю відсутній. На (рис.3.15) можна побачити що плагін бачить трекери так як чоловічок зеленого коляру.

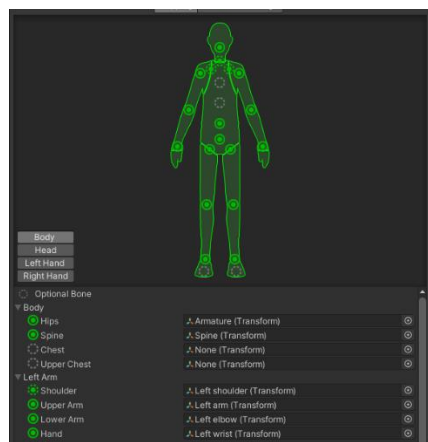


Рисунок 3.15 — Скелет VR аватара

Коли усі попередні кроки було виконано настав час перевірити роботу у віртуальному просторі, для цього по середині робочого вікна рушія Unity нажимаємо на `play` див. (рис 3.16.)

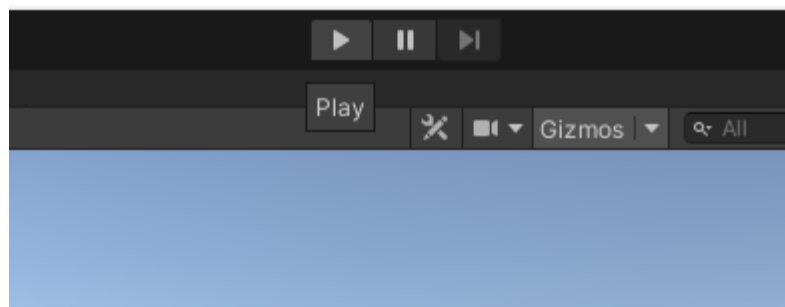


Рисунок 3.16 — Запустити сцену

					КС КРБ 123.023.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

Важливо зазначити що потрібно переконатися що програма яка була описана в пункті 3.1 запущена а шолом віртуальної реальності підключений — до ПК. Після того як усе в вище перерахованого було виконано, ми нарешті можемо побачити роботу трекерів у віар. Ми появимося в віар просторі в вбудованому аватарі який буде стояти у Т-позі. Це означає що потрібно провести калібрування трекерів. В віар трекери під час калібрування буду показуватися як невеликі білі сфери. Все що потрібно зробити це встати в Т-позу в реальному житі так щоб ці сфери були як омога блище до кінцівок аватару після чого натиснути на тригери на VR-контролерах, приклад див. рис. 3.17.

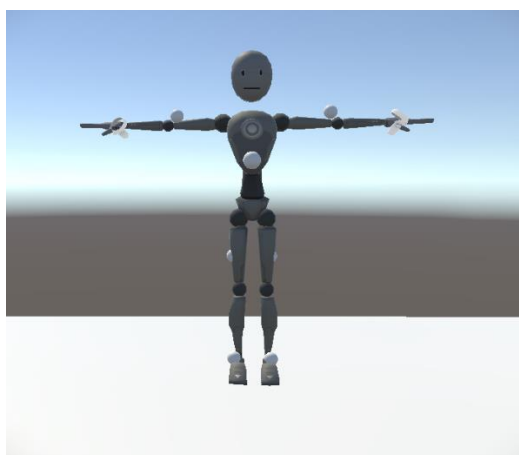


Рисунок 3.17 — калібрування трекерів.

Після калібрування трекери повині працювати, нижче наведено приклади їх роботи.

1) Приклад поза тіла, талію повернено в право (рис. 3.18, рис. 3.19).

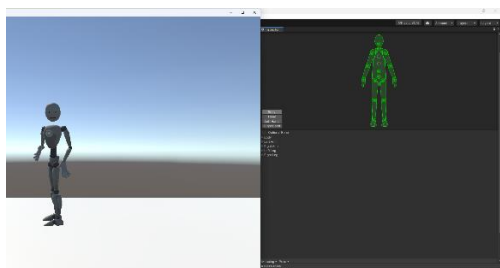


Рисунок 3.18 — Трекери увімкнено

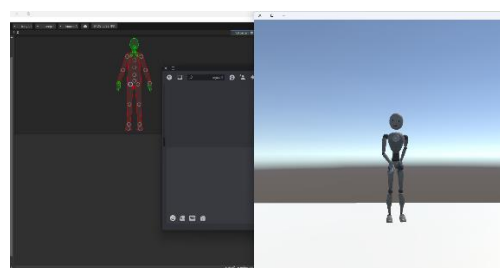


Рисунок 3.19 — Трекери вимкнено

					КС КРБ 123.023.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

2) Приклад поза тіла, бігун (рис. 3.20, рис. 3.21).

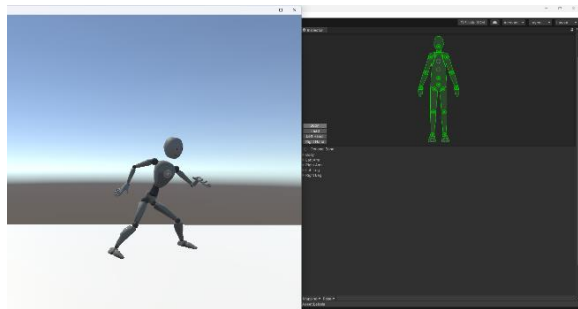


Рисунок 3.20 – Трекери увімкнено



Рисунок 3.21 – Трекери вимкнено

В результаті всіх проведених дій ми можемо побачити, що трекери працюють задовільно, не погано захоплюють та передають рухи людини на аватар у створеному нами віртуальному просторі на рушії Unity. Треки працюють автономно без додаткових станцій з камерами, вони працюють з даними які надходять від сенсорів гіроскопа та акселерометра, а прив'язка трекерів до Steam VR надає змогу допоміжного калібрування, тобто трекери крім отриманих даних від сенсорів щей отримують дані про положення контролерів та VR шолому в просторі що допомагає відтворювати положення тіла в випадку збою секундного роботи одного з трекерів.

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Техніка безпеки експлуатації гарнітури віртуальної реальності

При використанні гарнітури віртуальної реальності необхідно дотримуватися інструкцій з налаштування та експлуатації, що додаються до гарнітури віртуальної реальності [14]. Потрібно ознайомлення з рекомендаціями щодо апаратного та програмного забезпечення для використання гарнітури віртуальної реальності. Ризик виникнення дискомфорту може збільшитися, якщо не використовувати рекомендоване обладнання та програмне забезпечення.

Гарнітура віртуальної реальності та програмне забезпечення не призначені для використання з будь-якими несанкціонованими пристроями, аксесуарами та/або програмним забезпеченням. Використання несанкціонованих пристроїв, аксесуарів та/або програмного забезпечення може призвести до травмування користувача або інших осіб, спричинити проблеми з продуктивністю.

Щоб зменшити ризик дискомфорту, перед використанням гарнітури налаштовується фокус перегляду для кожного користувача.

Для комфортної роботи у віртуальній реальності необхідне нормальне відчуття руху та рівноваги. Не бажано використовувати гарнітуру віртуальної реальності, коли користувач втомлений; потребує сну; перебуває в стані емоційного стресу або тривоги; страждає від застуди, грипу, головного болю, мігрені або болю у вухах, оскільки це може підвищити сприйнятливність до несприятливих симптомів.

Змн.	Арк.	№ докум.	Підпис	Дата	КС КРБ 123.023.00.00 ПЗ			
Розроб.		Соколов О.М.			Безпека життєдіяльності, основи охорони праці.	Літ.	Арк..	Акрушів
Перевір.		Жаровський Р.О					48	57
Консультант		Пилипець М.І.				ТНТУ, каф. КС, гр. СІзс-		
Н. контр.		Тиш Є. В				42		
Затв.		Осухівська Г.М.						

Потрібно проконсультуватися з лікарем перед використанням гарнітури, якщо користувач вагітний, похилого віку, або має вже існуючі порушення бінокулярного зору або інші серйозні захворювання.

Щоб зарядити гарнітуру віртуальної реальності, потрібно обережно покласти її у футляр, що входить до комплекту постачання, і підключити USB-кабелю. Коли кейс підключено до мережі за допомогою кабелю живлення з комплекту постачання, на гарнітурі віртуальної реальності має загорітися індикатор, що вказує на те, що вона заряджається. Коли гарнітура повністю зарядиться, потрібно вимкнути живлення від мережі.

Дорослі повинні стежити за тим, щоб діти використовували гарнітуру відповідно до цих застережень щодо охорони здоров'я та безпеки, зокрема, щоб гарнітура використовувалася згідно інструкції [11].

Дорослі повинні стежити за дітьми, які користуються або користувалися гарнітурою, на предмет появи будь-яких симптомів, описаних у застереженнях щодо охорони здоров'я та безпеки, а також обмежити час, який діти проводять з гарнітурою, і забезпечити перерви під час її використання. Слід уникати тривалого використання, оскільки це може негативно вплинути на зорово-моторну координацію, рівновагу та здатність до багатозадачності. Дорослі повинні уважно спостерігати за дітьми під час і після використання гарнітури віртуальної реальності на предмет погіршення цих здібностей. Потрібно використовувати ClassVR не більше 15 хвилин на одному уроці. Короткі сеанси віртуальної реальності дуже захоплюючі і є ідеальним способом розпочати тему або закріпити ключовий момент. Плеєр ClassVR сповіщає вчителя, якщо учень перебуває у віртуальній реальності довше рекомендованого часу, що також є межею, рекомендованою для дітей оптометристами [9].

Щоб зменшити ризик травмування або дискомфорту, завжди потрібно дотримуватися інструкцій і запобіжних заходів під час використання гарнітури віртуальної реальності.

Необхідно використовувати тільки в безпечному середовищі: гарнітура віртуальної реальності створює ефект віртуальної реальності з ефектом

					КС КРБ 123.023.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

занурення, який відволікає і повністю блокує погляд користувача на реальне оточення.

Слід слідкувати за навколишнім оточенням перед початком використання та під час використання гарнітури віртуальної реальності. Необхідно бути обережним, щоб уникнути травм, і залишатися сидючи, якщо тільки перегляд контенту не вимагає стояння. Використання гарнітури віртуальної реальності може призвести до втрати рівноваги.

Необхідно пам'ятати, що об'єкти, які відображаються у віртуальному середовищі, не існують у реальному оточенні, тому не слід сідайти або стояти на них і не використовувати їх як опору.

Серйозні травми можна отримати, спіткнувшись, нашттовхнувшись або вдарившись об стіни, меблі чи інші об'єкти, тому перед використанням гарнітури потрібно розчистити ігрову зону для безпечного використання.

Потрібно бути особливо обережним, щоб не перебувати поблизу інших людей, предметів, сходів, балконів, відкритих дверей, вікон, меблів, відкритого вогню, стельових вентиляторів, світильників та інших предметів, з якими можна зіткнутися або вдаритися під час використання гарнітури віртуальної реальності або відразу після її використання.

Перед використанням гарнітури віртуальної реальності потрібно усунути всі небезпечні місця, де можна спіткнутися, перед тим, як користуватися нею.

Під час використання гарнітури користувач може не усвідомлювати, що в безпосередній близькості від нього можуть перебувати люди [9].

Заходи використання гарнітури віртуальної реальності:

— Під час використання гарнітури віртуальної реальності забороняється використання гострих або інших небезпечних предметів.

— Забороняється носити гарнітуру віртуальної реальності в ситуаціях, що вимагають уваги, під час ходьби, їзди на велосипеді або керування автомобілем.

					КС КРБ 123.023.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

— Слід впевнитися що гарнітура віртуальної реальності розташована рівно і зручно закріплена на голові, а також, що ви бачите єдине, чітке зображення.

— Потрібно, щоб кабелі навушників, якщо вони використовуються, не створювали небезпеки спіткнутися.

— Не слід поспішати використовувати гарнітуру віртуальної реальності, щоб дати можливість тілу пристосуватися; спочатку використовувати її лише кілька хвилин за раз, а потім поступово збільшуйте час використання гарнітури, коли користувач звикне до віртуальної реальності. Потрібно озирнутися навколо під час першого входу у віртуальну реальність, це допоможе пристосуватися до невеликих відмінностей між рухами в реальному світі та у віртуальній реальності.

— Забороняється використовувати гарнітуру, перебуваючи в транспортному засобі, що рухається, в автомобілі, автобусі або поїзді, оскільки це може підвищити сприйнятливість до несприятливих симптомів.

— Необхідно робити принаймні 10-15-хвилинну перерву кожні 30 хвилин, навіть якщо здається, що це не потрібно. Кожна людина індивідуальна, тому слід робити частіші та довші перерви, якщо користувач відчуватиме дискомфорт.

— Якщо використовувати навушники, прослуховування звуку на високій гучності може завдати непоправної шкоди слуху. Фоновий шум, а також постійний вплив високих рівнів гучності може призвести до того, що звуки здаватимуться тихішими, ніж вони є насправді.

— Через ефект занурення у віртуальну реальність забороняється використання гарнітури віртуальної реальності з високою гучністю звуку, щоб не втрачати відчуття оточення і зменшити ризик пошкодження слуху.

					КС КРБ 123.023.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

4.2 Долікарська допомога при контузіях

Під час війни складно назвати точну цифру людей, які зазнавали контузії, зокрема через те, що ця травма великою мірою стосується ураження нервової системи, яке не завжди одразу діагностують. За даними Національної академії медичних наук України протягом 2014-2016 років у зоні АТО контузії різного ступеня тяжкості зазнало близько 80% військових. Як виявити її ознаки, надати першу допомогу і чому навіть незначні на перший погляд ураження можуть мати серйозні наслідки в майбутньому.

Контузія це загальне ураження організму внаслідок різкого механічного впливу (повітряної, водяної чи звукової хвилі, удару об землю чи воду тощо), що не обов'язково супроводжується механічними ушкодженнями органів і тканин. Характерною ознакою контузії є непритомність (подекуди аж до коми). У легких випадках людина непритомніє на кілька хвилин, у тяжких на кілька днів і навіть місяців.

Найрозповсюдженішою є контузія головного мозку [12]. Наслідки контузії головного мозку є різноманітними — від тимчасової втрати слуху, зору, мовлення з подальшим частковим чи повним їхнім відновленням, до тяжких порушень психічної діяльності.

Після опритомнення відзначається сильний головний біль, нудота й блювання (незалежно від приймання їжі), запаморочення, особливо під час поворотів голови, амнезія, порушення слуху й мовлення. Найбільш різкі порушення слуху, до самої його втрати, можливі при баротравмах. Наслідками тяжкої контузії є швидка втомлюваність, що довго зберігається, погане самопочуття, підвищена дратівливість.

Виділяють три види контузії:

- легка заїкання, тремтіння або поколювання кінцівок, погіршення слуху;
- середньої тяжкості відсутня реакція зіниць на світло, повна глухота, частковий параліч кінцівок;

					КС КРБ 123.023.00.00 ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

— важка людина втрачає свідомість, має переривчасте дихання, судороми чи мимовільні рухи кінцівками, з її рота і вух може витікати кров.

Причому ризик розвитку тривалих симптомів контузії не пов'язаний зі ступенем тяжкості початкової травми. Навіть якщо здається, що травма несуттєва, ліпше звернутися по фахову медичну допомогу аби уникнути можливих ускладнень у майбутньому. До них належать порушення пам'яті, концентрації, погіршення сприйняття інформації.

Як розпізнати контузію

Основні симптоми, які можуть свідчити про контузію:

- втрата свідомості, яка може тривати від кількох хвилин до годин;
- сильний головний біль;
- прискорене серцебиття;
- нудота і блювота;
- втрата пам'яті;
- переломи кісток;
- ушкодження внутрішніх органів;
- будь-яка зміна психо-емоційного стану після травми.

Як допомогти контуженій людині.

Необхідно звільнити груди і комір людини від одягу, що стискає, і обережно перевернути на бік. При травмах голови можлива блювота або виділення слизу, крові чи спинномозкової рідини. Коли людина без свідомості лежить на боці, знижується ризик того, що вона захлеснеться.

Якщо з носа чи вух витікала кров, їх слід обережно очистити від згустків. Коли кровотеча тривала, до вух і носа потерпілого варто вкласти кульки з вати.

Негайно повідомити медиків про контузію, або самостійно транспортувати потерпілого до найближчого медпункту на носилках у положенні лежачи на боці.

Що категорично заборонено робити.

Давати пити або робити штучне дихання. Це пов'язано з тим, що у людини можуть бути розриви внутрішніх органів.

					КС КРБ 123.023.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

Чому контузія може бути небезпечною в тривалій перспективі.

Шкідливими є ураження головного мозку — після удару вибухової хвилі різні його частини рухаються з різною швидкістю. Це призводить до розривів відростків нейронів, які з'єднують між собою різні відділи мозку. Через деякий час пошкоджені нервові клітини гинуть.

У разі удару додатково може порушитися потік нейромедіаторів — молекул, за допомогою яких клітини мозку “спілкуються між собою”. У неконтрольовано великих кількостях ці речовини є токсичними і призводять до так званого вторинного ушкодження нейронів. Як наслідок, в ураженій ділянці формується запалення і набряк, що додатково погіршує кровопостачання ураженого відділу мозку і сприяє додатковій загибелі нейронів.

Симптоми залежать від локалізації пошкодженої ділянки мозку і проявляються у вигляді втрати слуху чи зору, постійних головних болів, порушення психо-емоційних процесів (апатія, дратівливість, підвищена збудливість). Нервові клітини відновлюються дуже повільно, тож контужена людина відчуває наслідки ще довго після травми. Тому важливо якнайшвидше звернутися по допомогу фахівців, адже це може зупинити подальше пошкодження головного чи спинного мозку.

					КС КРБ 123.023.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

ВИСНОВКИ

Під час виконання кваліфікаційної роботи було розглянуто різні види VR трекерів та методи їх відстеження в просторі. Було розроблено схему трекера VR на базі сенсора гіроскопа та акселерометра, було наведено три можливі варіанта трекерів які відрізняються ціною та якістю а саме, трекери на базі сенсора MPU6050, BMI160 та BNO085. Розглянули в деталях усі важливі компоненти VR трекера.

Також було розроблено плагін для програми Steam VR за допомогою якого стало можливим підключення DIY VR трекерів до ПК, їх налаштування, калібрування та їх прив'язки до Steam VR. Усі розроблені трекери є автономними, вони чудово працюють без додаткових станцій з камерами, та повністю готові для використання після встановлення усього важливого ПЗ, при умові що є наявний любий VR шолом.

					КС КРБ 123.023.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

ПЕРЕЛІК ПОСИЛАНЬ

1. Aukstakalnis, Steve. Practical augmented reality: a guide to the technologies, applications, and human factors for AR and VR.
2. "Oculus App Store Will Require Pre-Approvals, Comfort Ratings, Tax". TechCrunch. 12 June 2015. Retrieved 2020-11-06.
3. Bleser, Gabriele; Stricker, Didier (February 2009). "Advanced tracking through efficient image processing and visual–inertial sensor fusion". Computers & Graphics.
4. R. Holloway and A. Lastra, "Virtual Environments: A Survey of the Technology," cs.unc.edu.
5. Rasmussen, Loki; Basinger, Jay; Milanova, Mariofanna (March 2019). "Networking Consumer Systems to Provide a Development Environment for Inside-Out Marker-Less Tracking for Virtual Reality Headsets".
6. "How Oculus squeezed sophisticated tracking into pipsqueak hardware". TechCrunch. 22 August 2019. Retrieved 2020-11-06.
7. A quick guide to Degrees of Freedom in Virtual Reality". Kei Studios. 2018-02-12. Retrieved 2020-11-06.
8. "The Optimal Number of Axes for Motion Sensors". CEVA's Experts blog. 5 February 2019. Retrieved 8 September/
9. Virtual Reality Health & Safety URL: <https://www.classvr.com/policies/health-and-safety> (дата звернення 5.06.2023)
10. T. Mazuryk, Virtual Reality History, Applications, Technology and Future. Vienna, Austria: Vienna University of Technology, 1996.
11. R. Holloway and A. Lastra, "Virtual Environments: A Survey of the Technology," cs.unc.edu. [Online] URL: <http://www.cs.unc.edu/techreports/93-033.pdf>. 2023р. (дата звернення 18.05.2023)
12. Shell Shock the Same as PTSD? They are the same—and they are different Stephen Joseph Ph.D. (Листопад 20, 2011).

					КС КРБ 123.023.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

13. "How virtual reality positional tracking works". VentureBeat. 2019-05-05. Retrieved 2020-11-06.

14. "The Optimal Number of Axes for Motion Sensors". CEVA's Experts blog. 5 February 2019. Retrieved 8 September 2022.

15. Jannick P. Rolland, Yohan Baillot, and Alexei A. Goon. A Survey of Tracking Technology for Virtual Environments.

16. Vikas Kumar N. Integration of Inertial Navigation System and Global Positioning System Using Kalman Filtering.

17. J. D. Hol, T. B. Schon, F. Gustafsson, P. J. Slycke. Sensor Fusion for Augmented Reality.

					КС КРБ 123.023.00.00 ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток А

Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

«Затверджую»

завідувач кафедри КС

_____ Осухівська Г.М.

" ____ " _____ 2023 р.

КОМП'ЮТЕРНА СИСТЕМА ВІДТВОРЕННЯ РУХІВ ЛЮДИНИ У
ВІРТУАЛЬНОМУ ПРОСТОРИ НА БАЗІ ГІРОСКОПІВ

ТЕХНІЧНЕ ЗАВДАННЯ

на 5 листках

Вид робіт:

Кваліфікаційна робота

На здобуття освітнього ступеня «Бакалавр»

Спеціальність 123 «Комп'ютерна інженерія»

«УЗГОДЖЕНО»

Керівник кваліфікаційної роботи

_____ к.т.н., ст. викл. Жаровський Р.О.

« ____ » _____ 2023 р.

«ВИКОНАВЕЦЬ»

Студент групи СІзс-42

_____ Соколов О.М.

« ____ » _____ 2023 р.

Тернопіль 2023

1. Загальні відомості

1.1. Повна назва та її умовне позначення.

Повна назва теми кваліфікаційної роботи «Комп'ютерна система відтворення рухів людини у віртуальному просторі на базі гіроскопів акселерометрів».

Умовне позначення кваліфікаційної роботи: КС КРБ 123.023.00.00

1.2. Виконавець.

Студент групи СІзс-42 факультету комп'ютерно-інформаційних систем і програмної інженерії, кафедри комп'ютерних систем та мереж, Тернопільського національного технічного університету імені Івана Пулюя, Соколов Олександр Михайлович.

1.3 Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи є наказ по університету (№4/7-180 від 11.05.2023 р.)

1.4 Планові терміни початку та завершення роботи

Плановий термін початку виконання кваліфікаційної роботи – 11.05.2023 р.

Плановий термін завершення виконання кваліфікаційної роботи – 14.06.2023 р

1.5 Планові терміни початку та завершення роботи

Порядок оформлення пояснювальної записки та графічного матеріалу здійснюється у відповідності до чинних норм та правил ІСО, ГОСТ, ЕСКД, ЕСПД та ДСТУ.

Пред'явлення проміжних результатів роботи з виконання кваліфікаційної роботи здійснюється у відповідності до графіку, затвердженого керівником роботи. Попередній захист кваліфікаційної роботи відбувається при готовності роботи на 90% , наявності пояснювальної записки та графічного матеріалу.

Пред'явлення результатів кваліфікаційної роботи відбувається шляхом захисту на відповідному засіданні ЕК, ілюстрацією основних досягнень за допомогою графічного матеріалу.

2. Призначення і цілі створення системи

2.1 Призначення і цілі створення системи

Система що розробляється призначена для захоплення та відтворення рухів людини у віртуальному просторі за допомогою датчиків на базі гіроскопа та акселерометра, без використання додаткових станцій з камерами.

2.2 Мета створення системи

Метою кваліфікаційної роботи є розробка Full body з не дорогих VR трекерів розробленими у цьому ж завданні які здатні працювати автономно, збираючи та надсилаючи дані про рух від вбудованих у них сенсорів. Система повина працювати з любым домашнім VR шоломом.

3. Склад виробу.

3.1. До складу виробу повинні входити:

- 1) Гіроском Акселерометр;
- 2) Wi-Fi модуль
- 4) Зарядний модуль;
- 5) комплект документації.

4. Технічні вимоги.

4.1. Вимоги по призначенню.

4.1.1. Трекери повинні мати наступні параметри:

1) Діапазон вимірювання кута повороту - ± 125 °/с: 262.4 LSB/°/с - ± 250 °/с:
131.2 LSB/°/с.

2) Діапазон вимірювання положення у просторі - ± 2 g: 16384 LSB/g - ± 4 g:
8192 LSB/g.

3) Години роботи від на зарядженому акумуляторі - 6 год. або більше.

4) Мінімальний ефект дрефту трекера.

4.1.2. Пристрій повинен живитись напругою від акумулятора 1200 мАг

4.2. Вимоги до умов експлуатації:

4.2.1. Температура експлуатації від 0 до +80°C

4.2.2. Відносна вологість до 100% при $t=25$ °C

4.3. Конструктивні вимоги.

4.3.1. Конструювання корпусу приладу в КІП не передбачено.

4.3.2. Для побудови системи має бути використана сучасна компонентна база.

4.3.3. При побудові системи необхідно передбачити наявність роз'ємів живлення.

4.3.4. Габаритні розміри при макетуванні, мм, не більше:

довжина 800

ширина 600

висота 600

5.4. Вимоги до надійності.

5.4.1. Система повинна відповідати вимогам ДСТУ 2862-94.

5.4.2. Наробка на відмову, не менше 5000 год.

6. Економічні показники.

6.1. Собівартість системи повинна бути не більше 3500грн.

7. Вимоги до документації.

7.1. Конструкторська документація повинна відповідати вимогам ЄСКД, ДСТУ та ГОСТ.

7.2. До складу документації повинно входити:

- 1) ПЗ
- 2) Структурна схема VR трекера
- 3) Перелік елементів
- 4) Блок схема алгоритму роботи

Додаток Б
Лістинг програмного забезпечення

1. Прошивка сенсора BNO08x:

```
#include <Adafruit_BNO08x.h>

#define BNO08X_CS 10
#define BNO08X_INT 9

// #define BNO08X_RESET 5
// but not for I2C or UART
#define BNO08X_RESET -1

Adafruit_BNO08x bno08x(BNO08X_RESET);
sh2_SensorValue_t sensorValue;

void setup(void) {
  Serial.begin(115200);
  while (!Serial) delay(10);

  Serial.println("Adafruit BNO08x test!");

  if (!bno08x.begin_I2C()) {
    if (!bno08x.begin_UART(&Serial1))
      if (!bno08x.begin_SPI(BNO08X_CS, BNO08X_INT)) {
        Serial.println("Failed to find BNO08x chip");
        while (1) { delay(10); }
      }
  }
  Serial.println("BNO08x Found!");
```



```

for (int n = 0; n < bno08x.prodIds.numEntries; n++) {
  Serial.print("Part ");
  Serial.print(bno08x.prodIds.entry[n].swPartNumber);
  Serial.print(": Version :");
  Serial.print(bno08x.prodIds.entry[n].swVersionMajor);
  Serial.print(".");
  Serial.print(bno08x.prodIds.entry[n].swVersionMinor);
  Serial.print(".");
  Serial.print(bno08x.prodIds.entry[n].swVersionPatch);
  Serial.print(" Build ");
  Serial.println(bno08x.prodIds.entry[n].swBuildNumber);
}

```

```

setReports();

```

```

Serial.println("Reading events");
delay(100);
}

```

Тут вказується, які виходи датчиків нам потрібно отримувати, у моєму випадку, потрібно усе:

```

void setReports(void) {
  Serial.println("Setting desired reports");
  if (! bno08x.enableReport(SH2_GAME_ROTATION_VECTOR)) {
    Serial.println("Could not enable game vector");
  }
}

void loop() {

```

```

delay(10);

if (bno08x.wasReset()) {
  Serial.print("sensor was reset ");
  setReports();
}

if (! bno08x.getSensorEvent(&sensorValue)) {
  return;
}

switch (sensorValue.sensorId) {

case SH2_GAME_ROTATION_VECTOR:
  Serial.print("Game Rotation Vector - r: ");
  Serial.print(sensorValue.un.gameRotationVector.real);
  Serial.print(" i: ");
  Serial.print(sensorValue.un.gameRotationVector.i);
  Serial.print(" j: ");
  Serial.print(sensorValue.un.gameRotationVector.j);
  Serial.print(" k: ");
  Serial.println(sensorValue.un.gameRotationVector.k);
  break;
}

}

```

2 Прошивка Wi-Fi модуля ESP8266:

```
#include "ESP8266WiFi.h"
```

```

#include "WiFiClient.h"

void setup() {
  Serial.begin(115200);

}

void loop() {
  WiFi.mode( WIFI_OFF );
  WiFi.forceSleepBegin();
  Serial.println("WiFi is down");
  delay(20000);

  WiFi.forceSleepWake(WIFI_ON);
  delay(1);
  Тут вмикається Wi-Fi
  WiFi.mode(WIFI_STA);
  WiFi.begin("163", "812");

```

Коли з'єднання було встановлено далі потрібно налагодити та установити IP Адреси

```

while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("WiFi connected");
Serial.print("IP address:5.58.201.243");

```

```
Serial.println(WiFi.localIP());
delay(2000);
}
```

Код плагіну

```
#![deny(clippy::all)]

use iced::{
    executor,
    theme::{self, Theme},
    time,
    widget::{
        button, canvas, checkbox, container, horizontal_space, scrollable, slider,
text,
        text_input, Column, Container, Row, Scrollable, Svg,
    },
    window, Alignment, Application, Color, Command, Element, Font, Length,
Settings, Subscription,
};

use circle::circle;
use iced_aw::Grid;
use joycon::{Battery, DeviceStatus, ServerStatus};
use needle::Needle;
use settings::WranglerSettings;
use std::{
    io::{
        self,
```

```

    prelude::{Read, Write},
  },
  net::SocketAddr,
  time::{Duration, Instant},
};
mod joycon;
mod steam_blacklist;
use steam_blacklist as blacklist;
mod circle;
mod needle;
mod settings;
mod style;
mod update;

const WINDOW_SIZE: (u32, u32) = (980, 700);

pub const ICONS: Font = Font::External {
  name: "Icons",
  bytes: include_bytes!("../assets/icons.ttf"),
};
pub const ICON: &[u8; 16384] = include_bytes!("../assets/icon_64.rgba8");

pub fn main() -> iced::Result {
  /*
  Let rgba8 =
image_rs::io::Reader::open("assets/icon.png").unwrap().decode().unwrap().to_rgba8()
;
  std::fs::write("assets/icon_64.rgba8", rgba8.into_raw());
  */
}

```

```

let settings = Settings {
  window: window::Settings {
    min_size: Some(WINDOW_SIZE),
    size: WINDOW_SIZE,
    icon: window::icon::from_rgba(ICON.to_vec(), 64, 64).ok(),
    ..window::Settings::default()
  },
  antialiasing: true,
  ..Settings::default()
};

match MainState::run(settings) {
  Ok(a) => Ok(a),
  Err(e) => {
    println!("{e:?}");
    print!("Press enter to continue...");
    io::stdout().flush().unwrap();
    let _ = io::stdin().read(&mut [0u8]).unwrap();
    Err(e)
  }
}
}

```

```

#[derive(Debug, Clone)]
enum Message {
  SettingsPressed,
  Tick(Instant),
  Dot(Instant),
  AddressChange(String),
  UpdateFound(Option<String>),
}

```

```

UpdatePressed,
BlacklistChecked(blacklist::BlacklistResult),
BlacklistFixPressed,
JoyconRotate(String, bool),
JoyconScale(String, f64),
SettingsResetToggled(bool),
SettingsIdsToggled(bool),
}

```

```
#[derive(Default)]
```

```

struct MainState {
    joycon: Option<joycon::Wrapper>,
    joycon_boxes: JoyconBoxes,
    search_dots: usize,
    settings_show: bool,
    server_connected: ServerStatus,
    server_address: String,

    settings: settings::Handler,
    update_found: Option<String>,
    blacklist_info: blacklist::BlacklistResult,
}

```

```

impl Application for MainState {
    type Executor = executor::Default;
    type Flags = ();
    type Message = Message;
    type Theme = Theme;

    fn new(_: Self::Flags) -> (Self, Command<Self::Message>) {

```

```

let mut new = Self::default();
new.joycon = Some(joycon::Wrapper::new(new.settings.clone()));
new.server_address = format!("{}",
new.settings.load().get_socket_address());
    (
        new,
        Command::batch(vec![
            Command::perform(update::check_updates(),
Message::UpdateFound),
            Command::perform(blacklist::check_blacklist(),
Message::BlacklistChecked),
        ]),
    )
}

fn title(&self) -> String {
    "SlimeVR Wrangler".into()
}

fn theme(&self) -> Theme {
    Theme::Dark
}

fn update(&mut self, message: Message) -> Command<Self::Message> {
    match message {
        Message::SettingsPressed => {
            self.settings_show = !self.settings_show;
        }
        Message::Tick(_time) => {
            if let Some(ref ji) = self.joycon {

```



```

    if let Some(res) = ji.poll_status() {
        self.joycon_boxes.statuses = res;
    }
    if let Some(connected) = ji.poll_server() {
        self.server_connected = connected;
    }
}
}
Message::Dot(_time) => {
    self.search_dots = (self.search_dots + 1) % 4;
}
Message::AddressChange(value) => {
    self.settings.change(|ws| ws.address = value);
}
Message::UpdateFound(version) => {
    self.update_found = version;
}
Message::UpdatePressed => {
    self.update_found = None;
    update::update();
}
Message::BlacklistChecked(info) => {
    self.blacklist_info = info;
}
Message::BlacklistFixPressed => {
    self.blacklist_info =
        blacklist::BlacklistResult::info("Updating steam config file.....");
    return Command::perform(blacklist::update_blacklist(),
Message::BlacklistChecked);

```

```

    }
    Message::JoyconRotate(serial_number, direction) => {
        self.settings.change(|ws| {
            ws.joycon_rotation_add(serial_number, if direction { 90 } else { -
90 });
        });
    }
    Message::JoyconScale(serial_number, scale) => {
        self.settings
            .change(|ws| ws.joycon_scale_set(serial_number, scale));
    }
    Message::SettingsResetToggled(new) => {
        self.settings.change(|ws| ws.send_reset = new);
    }
    Message::SettingsIdsToggled(new) => {
        self.settings.change(|ws| ws.keep_ids = new);
    }
}
Command::none()
}

fn subscription(&self) -> Subscription<Message> {
    Subscription::batch(vec![
        time::every(Duration::from_millis(500)).map(Message::Dot),
        time::every(Duration::from_millis(50)).map(Message::Tick),
    ])
}

fn view(&self) -> Element<Message> {

```

```

let mut app = Column::new().push(top_bar(self.update_found.clone()));

if self.blacklist_info.visible() {
    app = app.push(blacklist_bar(&self.blacklist_info));
}

app.push(
    if self.settings_show {
        container(self.settings_screen()).padding(20)
    } else {
        container(self.joycon_screen())
    }
    .width(Length::Fill)
    .height(Length::Fill)
    .style(style::container_darker as for<'r> fn(&'r _) -> _),
)
.push(bottom_bar(
    self.server_connected,
    &".".repeat(self.search_dots),
    &self.server_address,
))
.into()
}
}

impl MainState {
    fn joycon_screen(&self) -> Scrollable<'_, Message> {
        let mut grid = Grid::with_column_width(320.0);
        for bax in self.joycon_boxes.view(&self.settings.load()) {

```

```

        grid.insert(container(bax).padding(10));
    }
    let list = Column::new().padding(10).width(Length::Fill).push(grid);

    let list = list.push(
        container(text(format!(
            "Searching for Joycon controllers{ }\n\
            Please pair controllers in the bluetooth \
            settings of Windows if they don't show up here.",
            ".".repeat(self.search_dots)
        )))
        .padding(10),
    );
    scrollable(list).height(Length::Fill)
}

fn settings_screen(&self) -> Column<'_, Message> {
    Column::new()
        .spacing(20)
        .push(address(&self.settings.load()).address))
        .push(checkbox(
            "Send yaw reset command to SlimeVR Server after B or UP button
press.",
            self.settings.load().send_reset,
            Message::SettingsResetToggled,
        ))
        .push(checkbox(
            "Save mounting location on server. Requires SlimeVR Server v0.6.1
or newer. Restart Wrangler after changing this.",
            self.settings.load().keep_ids,

```

```

        Message::SettingsIdsToggled,
    ))
}
}

fn address<'a>(input_value: &str) -> Column<'a, Message> {
    let address = text_input("127.0.0.1:6969", input_value)
        .on_input(Message::AddressChange)
        .width(Length::Fixed(300.0))
        .padding(10);

    let address_row = Row::new()
        .spacing(10)
        .align_items(Alignment::Center)
        .push("SlimeVR Server address:")
        .push(address)
        .push("Restart Wrangler after changing this.");
    let mut allc = Column::new().push(address_row).spacing(10);

    if input_value.parse::<SocketAddr>().is_err() {
        allc = allc.push(
            container(text(
                "Address is not a valid ip with port number! Using default instead
(127.0.0.1:6969).",
            ))
            .style(style::text_yellow as for<'r> fn(&'r _) -> _),
        );
    }
    allc
}

```

```

}
fn top_bar<'a>(update: Option<String>) -> Container<'a, Message> {
    let mut top_column = Row::new()
        .align_items(Alignment::Center)
        .push(text("SlimeVR Wrangler").size(24));

    if let Some(u) = update {
        let update_btn = button(text("Update"))
            .style(theme::Button::Custom(Box::new(style::PrimaryButton)))
            .on_press(Message::UpdatePressed);
        top_column = top_column
            .push(horizontal_space(Length::Fixed(20.0)))
            .push(text(format!("New update found! Version: {u}. ")))
            .push(update_btn);
    }

    let settings = button(text("Settings"))
        .style(theme::Button::Custom(Box::new(style::PrimaryButton)))
        .on_press(Message::SettingsPressed);
    top_column = top_column
        .push(horizontal_space(Length::Fill))
        .push(settings);

    container(top_column)
        .width(Length::Fill)
        .padding(20)
        .style(style::container_highlight as for<'r> fn(&'r _) -> _)
}

```

```

fn blacklist_bar<'a>(result: &blacklist::BlacklistResult) -> Container<'a,
Message> {
    let mut row = Row::new()
        .align_items(Alignment::Center)
        .push(text(result.info.clone()))
        .push(horizontal_space(Length::Fixed(20.0)));
    if result.fix_button {
        row = row.push(
            button(text("Fix blacklist"))
                .style(theme::Button::Custom(Box::new(style::PrimaryButton)))
                .on_press(Message::BlacklistFixPressed),
        );
    }
    container(row)
        .width(Length::Fill)
        .padding(20)
        .style(style::container_info as for<'r> fn(&'r _) -> _)
}

```

```

fn bottom_bar<'a>(
    connected: ServerStatus,
    search_dots: &String,
    address: &String,
) -> Container<'a, Message> {
    let status = Row::new()
        .push(text("Connection to SlimeVR Server: "))
        .push(container(text(format!("{}", connected))).style(
            if connected == ServerStatus::Connected {
                style::text_green
            }
        ))
}

```

```

        } else {
            style::text_yellow
        },
    ))
    .push(text(if connected == ServerStatus::Connected {
        format!(" to {address}.")
    } else {
        format!(". Trying to connect to {address}{search_dots}")
    }));
container(status)
    .width(Length::Fill)
    .padding(20)
    .style(style::container_info as for<r> fn(&'r _) -> _)
}

```

```
#[derive(Debug)]
```

```

struct JoyconBoxes {
    pub statuses: Vec<joycon::Status>,
    svg_handler: joycon::Svg,
    needles: Vec<Needle>,
}

```

```
impl Default for JoyconBoxes {
```

```

    fn default() -> Self {
        Self {
            statuses: vec![],
            svg_handler: joycon::Svg::new(),
            needles: (0..360).map(Needle::new).collect(),
        }
    }
}

```



```

    }
}

impl JoyconBoxes {
    fn view<'a>(&'a self, settings: &WranglerSettings) -> Vec<Container<'a,
Message>> {
    self.statuses
        .iter()
        .map(|status| {
            container(single_box_view(
                status,
                &self.svg_handler,
                &self.needles,
                settings.joycon_scale_get(&status.serial_number),
                settings.joycon_rotation_get(&status.serial_number),
            ))
                .height(Length::Fixed(335.0))
                .width(Length::Fixed(300.0))
                .padding(10)
                .style(style::item_normal as for<'r> fn(&'r _) -> _)
            })
        .collect()
    }
}

fn single_box_view<'a>(
    status: &joycon::Status,
    svg_handler: &joycon::Svg,
    needles: &'a [Needle],

```

```

scale: f64,
mount_rot: i32,
) -> Column<'a, Message> {
  let sn = status.serial_number.clone();

  let buttons = Row::new()
    .spacing(10)
    .push(
      button(text("↶").font(ICONS))
        .on_press(Message::JoyconRotate(sn.clone(), false))
        .style(theme::Button::Custom(Box::new(style::PrimaryButton))),
    )
    .push(
      button(text("↷").font(ICONS))
        .on_press(Message::JoyconRotate(sn.clone(), true))
        .style(theme::Button::Custom(Box::new(style::PrimaryButton))),
    );

  let svg = Svg::new(svg_handler.get(&status.design, mount_rot));

  let left = Column::new()
    .spacing(10)
    .align_items(Alignment::Center)
    .push(buttons)
    .push(svg)
    .width(Length::Fixed(130.0));

  let rot = status.rotation;
  let values = Row::with_children(

```

```

[("Roll", rot.0), ("Pitch", rot.1), ("Yaw", -rot.2)]
  .iter()
  .map(|(name, val)| {
    let ival = (*val as i32).rem_euclid(360) as usize;
    let needle = needles.get(ival).unwrap_or_else(|| &needles[0]);

    Column::new()
      .push(text(name))
      .push(
        canvas(needle)
          .width(Length::Fixed(25.0))
          .height(Length::Fixed(25.0)),
      )
      .push(text(format!("{ival}")))
      .spacing(10)
      .align_items(Alignment::Center)
      .width(Length::Fill)
      .into()
  })
  .collect(),
);

let circle = circle(
  8.0,
  match status.status {
    DeviceStatus::Disconnected | DeviceStatus::NoIMU =>
Color::from_rgb8(0xff, 0x38, 0x4A),
    DeviceStatus::LaggyIMU => Color::from_rgb8(0xff, 0xe3, 0x3c),
    DeviceStatus::Healthy => Color::from_rgb8(0x3d, 0xff, 0x81),
  }
);

```

```
    },  
);
```

```
let top = Row::new()  
    .spacing(5)  
    .push(circle)  
    .push(left)  
    .push(values)  
    .height(Length::Fixed(150.0));
```

```
let battery_text =  
    container(text(format!("{:?}", status.battery))).style(match status.battery {  
        Battery::Empty | Battery::Critical => style::text_orange,  
        Battery::Low => style::text_yellow,  
        Battery::Medium | Battery::Full => style::text_green,  
    });
```

```
let status_text = container(text(format!("{}", status.status))).style(match  
status.status {  
    DeviceStatus::Disconnected | DeviceStatus::NoIMU => style::text_orange,  
    DeviceStatus::LaggyIMU => style::text_yellow,  
    DeviceStatus::Healthy => style::text_green,  
});
```

```
let bottom = Column::new()  
    .spacing(10)  
    .push(  
        slider(0.8..=1.2, scale, move |c| {  
            Message::JoyconScale(sn.clone(), c)        })
```

```

    })
    .step(0.001),
)
.push(text(format!("Rotation scale ratio: {scale:.3}")))
.push(
    text(
        "Change this if the tracker in vr moves less or more than your irl
joycon. Higher value = more movement.",
    )
    .size(14),
)
.push(Row::new().push(text("Battery level: ")).push(battery_text))
.push(Row::new().push(text("Status: ")).push(status_text));

Column::new().spacing(10).push(top).push(bottom)
}

```