

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВОДНОГО ГОСПОДАРСТВА ТА**  
**ПРИРОДОКОРИСТУВАННЯ**

“До захисту допущена”

Зав. кафедри комп'ютерних наук та прикладної математики

д.т.н., професор Турбал Ю. В.

«\_\_\_» \_\_\_\_\_ 2023 р.

**КВАЛІФІКАЦІЙНА РОБОТА**

«Проектування системи вимірювання рухів руки на основі акселерометрів та  
тензорезисторів»

**Виконав:** Голюк Костянтин Васильович

студент навчально-наукового інституту автоматичної, кібернетичної та  
обчислювальної техніки

група ПЗ-41

\_\_\_\_\_  
(підпис)

**Керівник:** доц., к.т.н. Жуковський Віктор Володимирович

\_\_\_\_\_  
(підпис)

Рівне – 2023

## ЗМІСТ

РЕФЕРАТ.....	4
ВСТУП.....	5
РОЗДІЛ I. СФЕРИ ЗАСТОСУВАННЯ ТА ПРЕДМЕТНА ОБЛАСТЬ.....	7
1.1. Що таке прилади зчитування рухів.....	7
1.2. Застосування приладів зчитування рухів.....	8
РОЗДІЛ II. ПІДХОДИ ДО ЗЧИТУВАННЯ РУХІВ.....	10
2.1. З використанням костюмів або їх складових.....	10
2.2. За допомогою камер.....	11
2.3. Змішаний підхід.....	12
РОЗДІЛ III. КОМПЛЕКТУЮЧІ ДЛЯ ЗЧИТУВАННЯ РУХІВ.....	14
3.1. Акселерометри.....	14
3.2. Гіроскопи.....	17
3.3. Тензорезистори та їх аналоги.....	19
РОЗДІЛ IV. ПРАКТИЧНА ЧАСТИНА.....	23
4.1. Теоретичні відомості.....	23
4.2. Фільтр Калмана.....	25
4.3. Проектування системи.....	29
4.4. Реалізація проекту.....	33
4.4.1. Прототипування.....	34
4.4.2. Програмування мікроконтролера.....	35
4.4.3. Програмування драйвера.....	37
РОЗДІЛ V. РЕКОМЕНДАЦІЇ ЩОДО ВИКОРИСТАННЯ.....	42
5.1. Інструкція користувача.....	42
5.2. Тестування людино-машинної взаємодії.....	44
ВИСНОВКИ.....	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	51

ДОДАТКИ.....	53
--------------	----

## РЕФЕРАТ

**Кваліфікаційна робота:** 55 ст., 27 рисунків, 1 додаток, 10 посилань.

**Мета роботи:** розробити пристрій (рукавицю) для зчитування жестів руки. Сконструювати просте та доступне рішення. Запрограмувати драйвер для роботи з ним і відобразити результат.

**Засоби розробки:** мова програмування C++. Середовище програмування мікроконтролера – Arduino IDE. Середовище програмування на мовах C – Visual Studio. Графічна бібліотека OpenGL для обробки та зображення даних. Графічний інтерфейс сумісний з OpenGL – ImGui. Програма для роботи з даними ком порту та виявлення критичних помилок – Hercules 3.2.4. Бібліотеки доповнення для OpenGL – GLAD, GLEW, GLM, STB.

**Актуальність роботи:** обумовлена швидким стрибком мехатронних систем та систем доповненої реальності. Також малою насиченістю ринку та потребою в більш доступних, простих рішеннях.

**Ключові слова:** доповнена реальність, мехатронні системи, системи зчитування рухів, рукавиці, розумні рукавиці, контролери, віртуальна реальність, COM, C++, акселерометри, тензорезистори.

## ВСТУП

Технологічний розвиток за останні роки, почав швидко рухатися в напрямку доповненої реальності та робототехніки. Доля компаній які займаються розвитком штучного інтелекту та систем управління або зчитування рухів, зросла. Це дозволило пришвидшити розвиток технологій, зокрема технологій захоплення рухів.

Серед великої кількості компаній на ринку технологій, на даний момент, існує жорстка конкуренція. Кожна з них намагається створити максимально просту і в той самий час ефективну систему зчитування рухів. Окремі гіганти інвестують в розвиток доповненої реальності, тому однією з вимог є наявність тактильних відчуттів. Це дозволило б передавати форму об'єкта та обмежувати рухи. Однією з таких компаній є Facebook Technologies. Ця компанія просуває свій власний бренд - Oculus. Більшість користувачів їх продукції очікують нових технологій які були неодноразово продемонстровані на різних технологічних виставках в яких брала участь компанія. Окремо, варто виділити компанію Microsoft що займається розробкою цілого ряду систем призначених для віртуальної реальності і на даний момент є головним постачальником систем віртуальної реальності в армії США.

Якщо дистанціюватися від фінансової частини питання, то можна зрозуміти важливість подібних приладів. Однією з причин, чому варто розвивати технології зчитування рухів – це для полегшення управління. Саме з цією метою на даний момент різні країни, інтегрують подібні системи в свої збройні сили та рятувальні служби. Такими країнами є: США, Німеччина, Франція, Італія, Британія та інші. Існування таких систем полегшує керування дронами та різними маніпуляторами, зокрема – саперними пристроями розмінування.

В Україні, використання приладів зчитування рухів руки, є життєво важливим. Вони дозволять полегшити роботу операторів бойових дронів та саперів. На жаль, під час бойових дій часто трапляються ситуації в яких недостатньо штатних систем керування. Багато військових майструють свої

прилади для полегшення роботи. Дуже відомим серед звичайних геймерів є використання маніпуляторів Steam Deck для управління стаціонарним вогневим комплексом “Шабля”.

Це є хорошим прикладом недостатності штатних систем управління або їх не достатньої зручності для керування. Саме для уникнення таких ситуацій, особливо під час бойових дій, існують різні пристрої зчитування рухів.

В сфері робототехніки теж досить популярне використання приладів зчитування рухів. Якщо говорити не про робото-технічні маніпулятори для виробництва то можна поглянути на людиноподібних роботів. Такі роботи дуже популярні в Японії. Є безліч різних фото і відео де зображені різні мехи що керуються людиною. Варто зазначити, що саме цей етап є проміжним до повноцінних ШІ роботів та, з моєї точки зору, є обов’язковим для досягнення хороших результатів більш просунутих машин. В ШІ робототехнічних систем, часто використовують прилади зчитування рухів для навчання. Це дозволяє пришвидшити процес навчання та надати базові рухи та жести які ШІ зможе використовувати в якості початкових даних.

Незалежно із усього що написано вище, найпопулярнішою сферою застосування є сфера кіно-індустрії та ігрової індустрії. В цих двох сферах найчастіше використовуються прилади зчитування рухів тіла та жестів рук. Саме з цих сфер почався стрімкий розвиток таких систем, оскільки сфера розваг є неприбутковою серед всіх інших. Великі прибутки та вимога реалістичної графіки або міміки людей в кадрі – дозволили розвиватися малим студіям що надають послуги зчитування рухів або графіки. Кожна з таких маленьких команд, вносить свій вклад в розвиток технологій для збільшення точності або здешевлення всієї системи в цілому. Багато з таких напрацювань беруться на озброєннями гігантами індустрій. Такими як Disney, Pixar, Warner Bros, Bethesda та інші.

# РОЗДІЛ І. СФЕРИ ЗАСТОСУВАННЯ ТА ПРЕДМЕТНА ОБЛАСТЬ

## 1.1. Що таке прилади зчитування рухів

Прилади зчитування рухів – це прилади, за допомогою яких можливе перетворення руху в цифровий сигнал який обробляється системою для подальшого використання. Якщо говорити більш простою мовою то можна сказати, що це прилад або прилади, які дозволяють зчитати рух і виконати певну дію. Яскравими прикладами таких приладів є різного роду костюми, рукавиці, системи камер і так далі. Також до них можна віднести контролери віртуальної реальності, джойстики та геймпади. Щоправда, джойстики та геймпади це прилади-маніпулятори і хоч вони дозволяють зчитувати рухи, це відбувається дуже обмежено і використання таких пристроїв в високоточній системі – неможливе.

Більшість з людей які чують про зчитування рухів, уявляють собі чорні костюми з білими шариками на одязі та обличчі. І дійсно – це так. Існує велика кількість систем які побудовані на використанні камер та цифрової обробки зображень. Проте здебільшого, в якості точок на одязі виступають датчики а сам одяг може нічим не відрізнитися від звичайного смокінгу чи спортивного костюму.

Здебільшого до приладів зчитування рухів відносять різні пристрої що працюють в єдиній системі та є допоміжними одна для одної. Залежність пристроїв проявляється в програмних обрахунках з урахуванням пристроїв попередників (більше інформації можна дізнатися в розділі 4 “підходи до зчитування рухів”). Це працює саме так тому що комплектуючі які використовуються, зазвичай, мало спілкуються між собою. Для прикладу розглядаючи акселерометри-гіроскопи можна зрозуміти, що два таких прилади будуть показувати положення відносно орієнтації в просторі а не один відносно іншого.

## 1.2. Застосування приладів зчитування рухів

Застосування пристроїв зчитування рухів є обширним і включає в себе багато сфер людської діяльності. Від звичайного відображення руху в цифровому вигляді до пілотування космічними апаратами на інших планетах. Єдина проблема яка обмежує роботу приладів – це швидкість передачі інформації та її точність. Саме з урахуванням такого недоліку проектуються майже всі просунуті системи.

У вступній частині можна коротко ознайомитися зі сферами застосування, тут ми розглянемо інші але не менш важливі сфери, для прикладу – сфера науки та освіти.

Сфера освіти потребує таких приладів не менше ніж розважальна та військова. Багато спеціалістів в різних галузях навчаються на тренажерах. Найцікавішими тренажерами для освіти є тренажери хірургії та пілотування. Багато хірургів навчаються проводити операції на спеціальних манекенах, але зі збільшенням складності операційних втручань, точність стає на перший план. В передових країнах де розвиток медицини на найвищому рівні вже не один рік використовуються маніпулятори. Звичайно ніхто не дасть проводити операцію від якої залежить життя людини тому, хто не отримав достатньої кількості теоретичних знань та практичних навичок. Знання студенти отримують під час навчань а практичні навички під час використання тренажерів та маніпуляторів. Саме тут і використовуються системи зчитування рухів. Вони дозволяють навчити майбутнього хірурга як проводити операції при цьому не наражаючи на небезпеку помилки живих людей. Теж саме можна побачити в пілотуванні.

Пілотування літака це складний процес який потребує феноменальної пам'яті та практичних умінь. Для більшості початкових пілотів навіть перший зліт є складним випробуванням. В кабіні пілота багато важливих приладів та кнопок, взаємодія з якими виконується в строго зазначеному порядку. При цьому, вартість літака може складати мільйони доларів і це без врахування життя екіпажу – воно є безцінним. Тому вартість помилки дуже висока. Для

того, щоб уникнути жертв, використовують прилади зчитування рухів та симулятори пілотування.

Ці дві сфери використання що наведені вище, можуть продемонструвати важливість теми та ціну розвитку без таких систем. Всього сфер де застосовуються тренажери та симулятори – величезна кількість. Варто враховувати те, що більшість напрямків розвитку в яких рухається світ, повністю або частково пов'язані з віртуальною реальністю та доповненою реальністю. В зв'язку з цим, актуальність таких систем буде зростати ще велику кількість років.

З моєї точки зору, найбільш популярною в використанні подібних систем у майбутньому, буде – підвищення спроможності людського тіла. Світ намагається розвивати тему екзоскелетів та систем які б підвищили показники людського організму. Якщо уявити систему яка реагує на рухи людини і відповідним чином покращує певну характеристику тіла: силу, спритність, висоту стрибка і т. д, то можна створити ідеальний екзоскелет, який дозволив би зменшити навантаження на людське тіло та зменшити кількість травмувань. Це було б корисно в професіях де вимагається використання людини. Таким чином, військові або шахтарі, змогли б вберегти себе і підвищити свою ефективність в рази. Така думка, використання систем зчитування рухів, з'являлась в багатьох компаніях що займаються розробкою екзоскелетів та людоподібних роботів. Навіть в деяких з них – це в повній мірі використовується.

## РОЗДІЛ II. ПІДХОДИ ДО ЗЧИТУВАННЯ РУХІВ

### 2.1. З використанням костюмів або їх складових

Досить популярний підхід до зчитування рухів тіла – за допомогою приладів користувача. Даний підхід з’явився з розвитком електронних комплектуючих. Перші системи які презентувалися, використовували саме цей підхід. Кожен любитель електроніки або професійний інженер, пам’ятає презентації компаній в яких вони показували різні окуляри віртуальної реальності та контролери положення рук.

Цей підхід потребує грамотного використання різних комплектуючих в повній, програмній, залежності одна від одної. Таким чином, з проектувати ефективну систему – складно, оскільки інженери-конструктори повинні враховувати цілий ряд особливостей роботи системи а інженери програмного забезпечення зобов’язані проводити зчитування показників і об’єднувати їх в повноцінну систему команд з урахуванням залежності приладів один від одного. Так що найбільша складність полягає в програмуванні та обробці команд з приладів.



**Рисунок 2.1.** Рукавиця доповненої реальності від Meta [10]

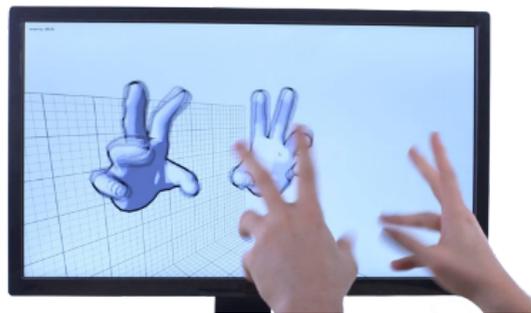
Цей підхід дозволяє створити просту та ефективну систему з компактними розмірами. Подібні системи на даний момент найпопулярніші в сфері доповненої реальності та віртуальної реальності. Одна з компаній яка займається розробкою саме такого роду систем є компанія Meta. Вони представили свою рукавицю доповненої реальності (рис. 2.1.) [10], щоправда рукавиця здатна відстежувати жести та рухи руки але не її положення в просторі. Для знаходження положення руки вони використовують камери. В

зв'язку з цим можна вважати саму рукавицю створену згідно даного підходу а систему в якій вона працює згідно змішаного.

Такий підхід є дуже популярним серед систем де не важливе положення руки в просторі а важливий сам жест. Для прикладу можна сказати, що така система зручна у використанні в якості джойстика або геймпаду.

## 2.2. За допомогою камер

Інший підхід до реалізації – це за допомогою однієї або багатьох камер. В більшості випадків він використовується досить рідко без першого. Це пов'язано з тим, що потребується камера і якщо вона одна то рух не є точним і більше схожий на приблизне уявлення руху. Так що, цей підхід потребує великої кількості камер розміщених навколо об'єкту рух якого зчитується.



**Рисунок 2.2.** Приклад системи зчитування рухів за допомогою камери користувача

В мережі інтернет можна зустріти багато варіантів реалізації такого підходу. Навіть є цілий ряд ресурсів які дозволяють використовувати камеру ноутбука для зчитування руху (рис 2.2.). Попри це, всі вони використовують ШІ. Останнім часом, його розвиток знаходиться в розквіті. Щороку з'являються нові напрацювання в сфері ШІ. Кожен спеціаліст для роботи якого необхідний комп'ютер, може знати такі ШІ як Chat GPT та Midjourney. Але вони мають більш вузьке призначення.

Особливості ШІ для опрацювання зображення та зчитування жесту на ньому, полягає в складності знаходження жесту у випадку якщо той закритий від камери. Можна уявити що користувач зігнув руку таким чином що не видно кисті. В такому випадку ШІ має робити висновок стосовно того, як виглядає

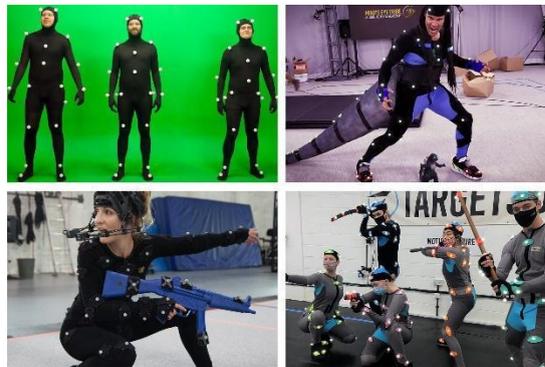
кисть руки та в якому положенні пальці. Якщо уникати цих сліпих зон то варто використовувати більше камер. В ту ж чергу це спричинить ускладнення роботи програмного коду. Для обрахунків варто буде враховувати позицію камер відносно одна одної та опрацювання фото в двох або більше, площинах.

Можна зробити висновок що така система не підходить для високоточного зчитування рухів. Більш доцільним є, використання системи для знаходження положення тіла в просторі.

Попри всі недоліки цього підходу, він часто використовується в простих системах де важливо зчитати положення руки на екрані і зараз він є популярним в різних спортивних програмах де він визначає правильність виконання вправи

### 2.3. Змішаний підхід

Найцікавішим підходом є змішаний. Чим він такий цікавий. В першу чергу варто зазначити що його використовують всі великі компанії в своїх приладах, бо він дозволяє знайти положення всіх частин тіла та жестів в реальному часі з великою точністю. Описати його використання допоможе найвідоміша варіація його застосування – motion capture.



**Рисунок 2.3.** Motion capture

Motion capture (рис. 2.3.) – це система камер та приладів користувача які направлені на зчитування рухів тіла в спеціально підготовленій площі. Якщо говорити коротко то це кімната в якій встановлена велика кількість камер та костюм який одягає актор. Такий спосіб зчитування найпопулярніший в ігровій та кіно індустрії. Якщо розглядати результати використання такого підходу то

най відомішим є фільм “Avatar” і мультфільм “Frozen II”. Саме в цих двох прикладах можна побачити найефективніше використання цієї технології.

Проблемою цього підходу є – велика складність програмування та використання. Є цілі великі команди з десятками а можливо і сотнями працівників що пишуть програмне забезпечення для таких систем. Більшість проблем виникає коли є необхідність зчитувати та обробляти інформацію з великої кількості камер та приладів користувача. Ще більше все ускладнюється якщо є потреба зчитати високоточно міміку обличчя людини. Ці всі складнощі є причиною великої вартості систем. Загальна вартість може варіюватися від декількох сотень тисяч доларів до мільйонів доларів, що унеможливує її використання в домашніх умовах та повністю зменшує доступність для звичайних користувачів.

В даний момент є багато різних розробок які дозволили б уникнути використання такої великої кількості камер та збільшити доступність систем. Більшість з них намагається використовувати прилади користувача по максимуму. В продажу є системи, вартість яких близько 5 тисяч доларів. Це досить припустима ціна за зручність та компактність.

## РОЗДІЛ III. КОМПЛЕКТУЮЧІ ДЛЯ ЗЧИТУВАННЯ РУХІВ

### 3.1. Акселерометри

На даний момент складно уявити наш світ без акселерометрів та гіроскопів. Кожен з нас користується такими приладами щоденно. Телефони, фітнес браслети та годинники – не можуть існувати без них. Щоправда, мало хто знає як вони працюють і навіщо використовуються.

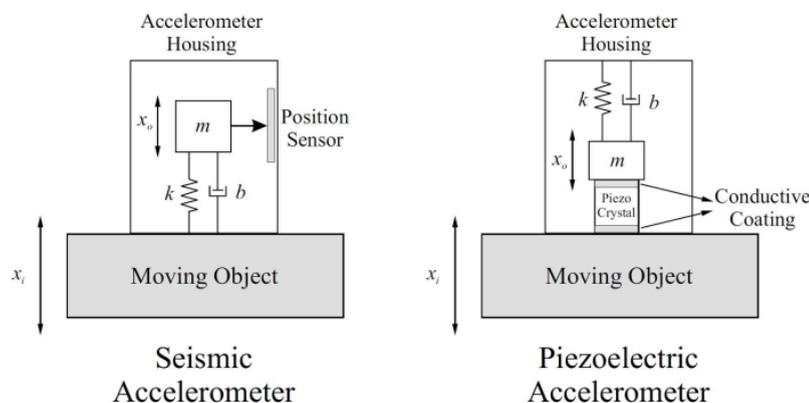
Акселерометр – це пристрій який вимірює зміну швидкості. Назва походить від “accelegero” що дослівно означає “прискорення” та “metrico”- “вимірювати”. Знаходження значення прискорення дозволяє виміряти з якою швидкістю та в якому напрямку рухається об’єкт. Завдяки цьому, фітнес браслет може зчитати кількість кроків та відстань що проходить користувач.

З конструктивної точки зору, акселерометри поділяються на однокомпонентні, двокомпонентні та трьох компонентні. Або, якщо говорити більш простіше то, одноосьові, двохосьові та трьохосьові. При чому, трьох осьові акселерометри здатні знаходити прискорення тіла в усіх трьох координатних площинах. Часто, акселерометри плутають з гіроскопами. Це пов’язано з тим, що більшість сучасних акселерометрів мають вбудований гіроскоп і для не обізнаного користувача, сплутати їх цілком нормальна практика. Варто розрізняти ці сенсори бо вони мають різну будову і виконують різні функції в конкретному приладі. В більшості випадків, акселерометр використовують для знаходження орієнтації в просторі, падінь, вібрацій та прискорення по координаті. Прикладом цього є телефон. Саме акселерометр визначає положення зображення на екрані.

Акселерометри використовують в різних пристроях, для прикладу:

- 1) Пристрої навігації кораблів, авіації, бпла. Без пристроїв на основі акселерометра не може існувати ні літака ні корабля. Завдяки ним вони можуть враховувати швидкості та напрям руху для стабілізації до рівня горизонту або в напрямку площини руху.

- 2) Автомобілі. В автомобільній сфері вони використовуються в системах безпеки та стабілізації. Екстрене гальмування або аварія зчитується приладом для того, щоб задіяти подушки безпеки.
- 3) Промисловість. Тут часто їх використовують для вимкнення пристроїв щоб уникнути нещасних випадків або критичних пошкоджень устаткування.
- 4) В ноутбуках і ПК. При падінні або сильному ударі, пристрій говорить жорсткому диску щоб той зайняв безпечне положення. Саме це часто стає причиною вимкнення системи комп'ютера під час падіння або сильного удару.

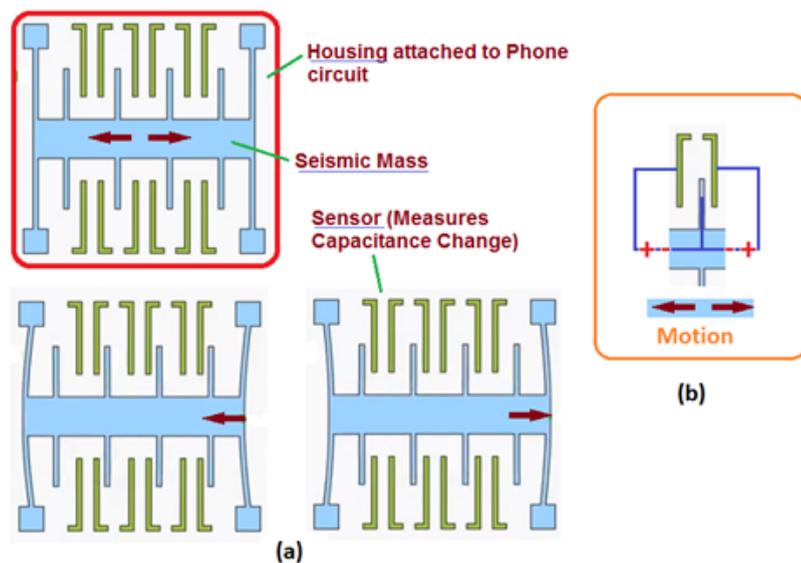


**Рисунок 3.1.** Приклад двох акселерометрів з різними принципами дії [1]

Відкритим питанням залишається те, як працює акселерометр. Більшість приладів мають ємнісний, п'єзоелектричний і п'єзорезистивний принципи дії. Більшість має будову MEMS (мікро-електронно-механічна система) яка складається з цілого ряду компонентів, розміри яких 1 – 100 мікроміліметрів. При цьому, сам пристрій може поміститися в невеличку мікросхему 2 на 2 на 1 міліметри.

Розглянемо його роботу на простому прикладі. На **рисунку 3.1**, можна побачити 2 найпростіші акселерометри. Хоч вони і мають різну будову але їх основна ідея однакова. Коли об'єкт на якому встановлений акселерометр рухається то створюється прискорення, яке спричиняє деформацію пружини. Пружина під дією маси (буква **m** на схемі вище) та прискорення тіла, починає

розтягуватися або стискатися. Це відбувається тому, що маса починає рухатись в протилежному напрямку від вектору напрямку руху. Таке можливе лиш тому, що маса не закріплена і може рухатися по прямій в одній площині. Знаючи те, наскільки сильно деформується пружина і скільки часу пройшло, можна визначити прискорення тіла в відповідному напрямку. Праворуч на фото вище, можна побачити п'єзоелектричний акселерометр. В ньому маса тисне на п'єзоелемент що виробляє електричний струм відповідно до сили тиску на елемент. З простим прикладом роботи акселерометру ми ознайомилися, проте зараз більшість систем використовують чипи акселерометри. Їх будова дещо відрізняється. Розглянемо її на прикладі одного з датчиків.



**Рисунок 3.2.** Приклад роботи SMD акселерометра

На рисунку вище, наведений принцип роботи SMD акселерометру. В ньому можна побачити рухому пластинку чітко вивіреної форми (блакитний колір) та зелені контакти, частина з яких (в деяких випадках усі) приєднані до ємнісних пластин. Таким чином, можна зрозуміти принцип його роботи. Він базується на зарядці певних конденсаторів та розрядці інших. При нерухомій пластині, ємнісний заряд конденсаторів однаковий. Як тільки пластина починає свій рух, її ніжки стають ближчими до контактів конденсаторів що знаходяться в напрямку руху пластини. Ті які ближче до ніжок отримують більше заряду,

віддалені – менше. Саме ця різниця зарядів ємнісних пластин (конденсаторів) і буде вказувати на напрям руху, а їх різниця - на прискорення.

Підбивши підсумки можна зазначити, що є багато різних способів реалізації та принципів роботи акселерометрів. Єдине що залишається незмінним – вимірювання відхилення та його напрямку.

### 3.2. Гіроскопи

Гіроскоп в час свого відкриття одразу став революцією в мореплавстві. Раніше морякам доводилося проводити величезну кількість розрахунків щоб знайти своє положення в просторі. Також це стосувалося багатьох інших спеціалістів які багато часу проводили в місцях де не було орієнтирів. До винайдення гіроскопа використовували рівень, астролябію, відвіс та інші. Лиш в 1817 році всі ці прилади можна було замінити 1 – гіроскопом.

Гіроскоп – це прилад здатний реагувати на зміну кутів орієнтації тіла на якому його було встановлено відповідно до початкової, інерційної системи відліку [2]. Нововинайдений пристрій має значну перевагу перед старими приладами. Найбільшою з них стала висока точність показників у складних умовах експлуатації.

В цілому, гіроскоп можна поділити на двохосьові та трьохосьові. Кількість осей говорить про те, в скількох площинах прилад може працювати (те, скільки він має ступенів свободи). В залежності від цього, його використовують в різних приладах де він буде експлуатуватися. Окрім цього, гіроскопи поділяються з принципом роботи, а саме: механічні, оптичні.

Механічні гіроскопи (рис. 3.3.) це гіроскопи які використовують механічний принцип роботи. Яскравим прикладом такого гіроскопу є роторний. Роторний гіроскоп – це тверде тіло що обертається на великій швидкості по осі яка може вільно змінювати орієнтацію в просторі. При чому, швидкість обертання значно більша за швидкість повороту його осі обертання. Головною перевагою такого гіроскопа є здатність зберігати незмінне направлення осі обертання при відсутності впливу зовнішніх сил. Ще однією перевагою є

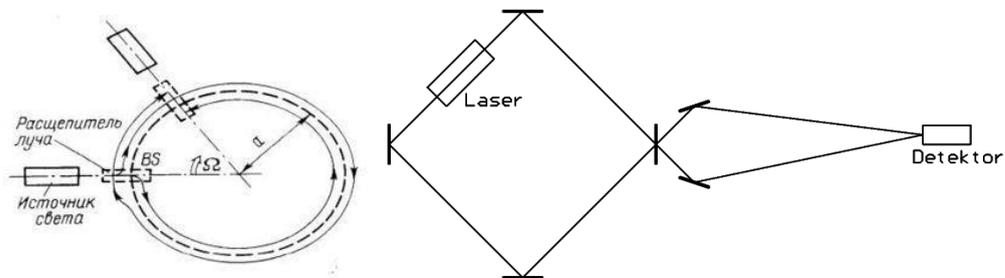
ефективний супротив зовнішнім силам. Ця властивість визначається величиною кутової швидкості обертання гіроскопа.



**Рисунок 3.3.** Механічний гіроскоп

Процесія механічного гіроскопа. Процесія виникає при дії моменту зовнішньої сили навколо осі, яка пролягає перпендикулярно осі обертання ротора. В цей момент, гіроскоп починає обертатися навколо осі процесії яка перпендикулярна моменту зовнішньої сили. Це обумовлено дією сили коріоліса.

Оптичні гіроскопи діляться на 2 види за принципом роботи: лазерні та волоконно-оптичні.



**Рисунок 3.4.** Принцип роботи волоконно-оптичного (зліва) та лазерного (з права) гіроскопів

Лазерні гіроскопи [3] зазвичай являють собою кільцевий резонатор з трьома або чотирма відбивачами розміщеними по кутам у формі трикутника або квадрату. Два лазерних промені, що були згенеровані в порожнині гіроскопа, безперервно циркулюють крізь резонатор в протилежних напрямках. У лазерному гіроскопі створюється та підтримується світлова хвиля, вузли якої ідеально пов'язані з інерційною системою відліку. Сам принцип роботи,

побудовані на ефекті Саньяка. При обертанні гіроскопа, генеруються хвилі різної частоти і різниця цих частот вказує на орієнтацію в просторі.

Волокнисто-оптичні гіроскопи теж базуються на ефекті Саньяка. В них промінь світла проходить через котушку оптоволокна. Зазвичай для підвищення точності та чутливості гіроскопа, використовують оптоволокно довшиною близькою до 1000 метрів вкладених в спіраль. Унікальним для даного гіроскопа є і те, що він використовує світло з малою хвилею когерентності, що необхідно для збільшення точності до необхідного та достатнього рівня.

Як можна зробити висновок – оптичні гіроскопи використовують здебільшого ефект Саньяка. При цьому вони мають різну будову і внутрішній механізм. Це не дозволяє помістити пристрій в малий по об'єму корпус, що унеможливило використання таких гіроскопів в компактних пристроях.

На даний момент, гіроскопи мають достатню точність і задовольняють потреби більшості користувачів. Якщо розглядати часовий інтервал між тим, коли гіроскопи стали доступними не військовим компаніям та простим користувачам і теперішнім часом, можна побачити тенденцію до розвитку мікросхематичних гіроскопів. Мікросхематичні гіроскопи це гіроскопи які знаходяться в компактному корпусі та мають високу точність. Зараз їх часто використовують в стабілізації камер та автомобілів. Для прикладу системи гідравлічної стабілізації авто, на пряму пов'язані з гіроскопами. Прошивка електроніки автомобіля отримує команду від гіроскопа яка змушує гідравліку виконувати дії для стабілізації. З моєї точки зору, саме мікросхематичні гіроскопи будуть мати найбільший розвиток в майбутньому.

### **3.3. Тензорезистори та їх аналоги**

Одна з найпоширеніших компонент це тензорезистори або ж – резистори деформації. Більшість з них побудовані на 1 ідеї а саме – вимірювання зміни опору. Розглянемо що таке тензорезистор та які існують їх види.

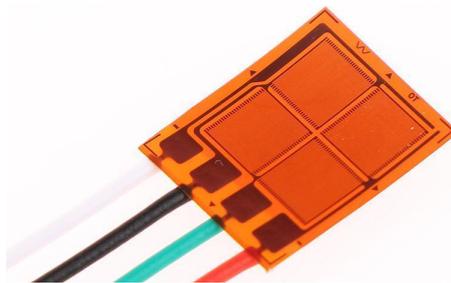
Тензорезистор – це сенсори що використовують властивість деформації матеріалів для знаходження зміни опору. В цілому, можна говорити що це

пристрої які при деформації змінюють свій опір або інший параметр. Їх часто поділяють на резистори тиску, згину та розтягу. Незалежно від цього, вони є резисторами деформації.

Більшість з нас часто стикається з використанням тензорезисторів тиску. Вони в свою чергу є дуже популярними в електронних вагах, завдяки своєму принципу дії. На відміну від резисторів згину, тут використовується зміна площі контакту за допомогою тиску а не розтягу або стискання. З цього і походить назва.

Резистори згину отримали свою популярність завдяки робототехніці та мехатроніці, оскільки вони дозволяють зчитувати положення одноосьових рухомих частин. Часто саме їх використовують для зчитування згину пальців.

Тензорезистори поділяються на 3 види: фольгові, плівкові, дротяні. Розглянемо їх детальніше.



**Рисунок 3.5.** Фольгований тензорезистор

Фольговані тензорезистори – це тонка металева стрічка, частина якої вибрана (методом травлення, лазером чи іншим високоточним способом) таким чином, що метал утворив металеву решітку з виходами для зчитування інформації. Металева решітка з обох боків покривається міцною ізолюючою плівкою. Це один з найпопулярніших видів тензорезисторів, завдяки малій вартості, габаритам та простоті монтажу. В порівнянні з іншими, вони мають кращі характеристики і дозволяють створювати решітку будь якої форми та розміру [4]. На відміну від дротяних, вони мають прямокутну решітку що при малій товщині дозволяє збільшити площу контакту з об'єктом вимірювання і цим самим – збільшити точність.



**Рисунок 3.6.** Плівковий тензорезистор

Іншим конструктивним видом тензорезисторів є плівкові. Плівкові тензорезистори дуже схожі до фольгових. Їх відмінністю є форма контактів та спосіб нанесення металу. Металеві доріжки наносяться методом вакуумної сублімації з подальшим осадом у відповідні місця на плівці. Фактично, процес нанесення доріжок виглядає дуже схожим чином до конденсації роси. Спочатку металеві частинки конденсуються на поверхню плівки, після чого – вони “приклеюються”. Це відбувається не на всій площині плівки а лише на тій її частині, яка покрита спеціальним покриттям що не дозволяє частинкам осипатися з нього. Прикладом чогось віддалено подібного є картинки які, більшість з нас, робили за допомогою нанесення клею на папір, після чого папір покривається сухою манкою. Та частина яка не потрапила на клей, з легкістю обсіпалася. Плівкові тензорезистори теж досить популярні. Їх роблять як з металів так і з напівпровідникових матеріалів, таких як графіт, германій, кремній і т. д.

Дротяні тензорезистори, мало чим відрізняються від плівкових чи фольгових. Головною відмінністю є те, що в їх основі лежить тонкий дріт. Також, вони відрізняються формою контактів. Фольгові мають форму решітки а плівкові – форму решітки чи паралельних доріжок. Дротяні можуть мати лише паралельні доріжки які послідовно з’єднані між собою. Ця особливість часто використовується в трубопроводах де немає необхідності в високоточних показниках але є потреба вимірювання деформації в одному напрямку.

В пристроях інтернету речей, тензорезистори використовуються надзвичайно часто. В більшості випадків саме в системах зчитування рухів. Їх вагомою перевагою перед акселерометрами та гіроскопами в випадку

зчитування згину, є надзвичайна простота монтажу і програмування. Саме їх можна побачити на різного роду рукавицях та костюмах у місцях згину. Тим не менш, є багато способів зробити саморобні тензорезистори. Така потреба часто виникає через велику вартість високоточних та професійних сенсорів. Можна розглядати 2 шляхи розвитку саморобних сенсорів.

Перший і найпопулярніший – оптичний. З назви стає зрозуміло що саме лежить в його основі. В більшості випадків такого роду сенсори виглядають як трубка з одного краю якою встановлений світлодіод а з іншого фоторезистор. Прилад виходить досить масивний але його простота роботи та дешева вартість може бути корисною якщо немає потреби в зчитуванні напрямку згину і високої точності. Недолі який є досить відчутний це часто переламані трубки. Дехто модифікує їх додаючи пружини, трубки враховуючи місце згину та простір для більш плавного кута.

Другий – це саморобний варіант звичайного тензорезистору. Він також використовує деформацію та опір напівпровідникового матеріалу. Виглядає він як бутерброд з фольги по краях та напівпровідника (здебільшого графіту) посередині. Загальний принцип його роботи такий самий як і в промислових, більша площа контакту – більший опір, менша площа – менший опір. Такий саморобний сенсор, потребує точного підбору матеріалів і не може гарантувати стабільну роботу. Хоча, часто його використовують через компактність та простоту.

Під час проведення особистих тестів різних саморобних тензорезисторів, можна сказати що оптичні є найточнішими та найнадійнішими. Вони показують стабільність навіть в відносно складних умовах експлуатації. Саме цей вид саморобних сенсорів буде використовуватися в практичній частині роботи. Також варто зазначити що саморобний фольгований сенсор при правильному підборі матеріалів і використанні ряду готових рішень, може теж мати хороший результат на виході. Єдина проблема яка часто виникає при його конструюванні – це погана графітова прокладка.

## РОЗДІЛ IV. ПРАКТИЧНА ЧАСТИНА

### 4.1. Теоретичні відомості

Перед початком переходу до практичного рішення, необхідно ознайомитися з деякими теоретичними відомостями направленними саме на практичне рішення.

Реалізувати систему зчитування рухів руки можна різними способами. Деякі з них потребують особливих навичок програмування і проектування а інші – є простими рішеннями які нам підійдуть оскільки метою є створення ефективної та максимально дешевої системи. Таким чином можна відкинути цілий ряд комплектуючих вартість яких є досить високою. До таких входять різні промислові рішення, в особливості промислові тензорезистори згину. Вони чудово підійшли б для нашої системи оскільки вони довгі (від 5 см.) та досить точні. При цьому, багато з них можуть працювати в двох напрямках що було б корисно. Все це нівелюється високою вартістю. Ціни на китайському ринку починаються від 500 гривень на 1 палець. Отже, при проектуванні нашої системи з подальшим прибиранням сліпих зон – вартість буде колосальною. Найкращий спосіб буде це використати звичайний саморобний оптичний тензорезистор.

Серед різновиду рішень було обрано тензорезистор на основі фоторезистору та світлодіоду. Опис приладу можна зустріти в розділі “тензорезистори”, там принцип його роботи розписаний досить чітко щоб можна було його відтворити. Окрім цього, варто враховувати те, що для коректної роботи нам необхідні будуть резистори щоб він працював стабільно. Звичайно, можна використовувати 1 резистор який буде резистивним подільником з фоторезистором але це потребуватиме окремої лінії живлення світлодіоду, що не завжди є хорошим рішенням.

Якщо говорити з теоретичної точки зору, то можна спроектувати таку систему на основі не трубки а світловоду. Така думка часто зустрічається на різних форумах де піднімається питання зчитування згину пальців. І дійсно, якщо уявити таку систему то можна дійсно дійти до висновку що

використовувати світловід це хороше рішення. Проте, з моєї точки зору, це може мати певний негативний ефект. В першу чергу це пов'язано з кількістю світла що проходить через світловід. Є певні підозри що при згині пристрою, кількість світла майже не змінюється а можливо і не зміниться зовсім. З іншого боку можна сказати що система буде дуже компактною та простою. Січення деяких світловодів близько 1 мм і при правильному підборі світлодіода, розмір системи не сильно зміниться що дозволить зробити її мега компактною. Єдина вимога це фоторезистор високої чутливості з малими габаритами. Чутливість необхідна для того, щоб він міг виміряти мінімальні зміни освітленості провідника. За подібним принципом працюють багато компаній. Це дозволяє їм використовувати малі габарити і досить хорошу точність в своїх продуктах.

З тензорезистором ми розібралися, тепер варто перейти до сенсору знаходження положення тіла в просторі. Для таких цілей найкраще підійде трьохосьовий акселерометр-гіроскоп MPU6050. Це просте, дешеве, і що важливо – дуже популярне рішення. Чому було сконцентровано увагу на його популярність. Це пов'язано з великою кількістю готових проектів та бібліотек для його роботи.

В перших спробах програмування виникли певні труднощі з обробкою інформації з MPU. Причина цього ховалася в спробах програмування на STM32. Як виявилось, програмування на STM32 вимагає величезної кількості знань комплектуючих. Така необхідність є завдяки “популярності” любительського програмування. Дуже мало різних рішень в інтернеті застосуванням MPU. Можна сказати що їх одиниці. Мала популярність рішень пов'язана з програмуванням складних проектів які приносять фінансову вигоду. Іншими словами – на їх програмний код лежить ліцензія яка не дозволяє заливати його в загальний доступ. Саме через це більшість починає з Arduino. Все ж, незважаючи на недоліки, була написана власна бібліотека для роботи з MPU. І тут виникла ще одна проблема. В зв'язку з використанням розширювача I2C портів, довелося писати ще одну бібліотеку для роботи з ним. На той час, інформації як працювати з ним на STM32 знайти не вдалося. Це стало головною

причиною переходу на Arduino. Бібліотека яка була написана для розширювача виявилась не сумісною з раніше написаною для MPU. Декілька днів роботи з datasheet комплектуючих та програмування на C – були витрачені впусту. Після переходу на Arduino, таких проблем не було. Завдяки високій популярності розробки на ATmega328P, кількість бібліотек та прикладів незчисленно велика. Це дозволило запрограмувати MPU та ТСА без проблем.

Дані гіроскопа та акселерометра пристрій відправляє сирими. Це означає що є потреба оптимізувати дані для отримання покращеного результату. Для того щоб це зробити необхідно мати уявлення того, як співпрацюють між собою гіроскоп та акселерометр. Гіроскоп в MPU завжди рухається, через це пряма яка з ним співвідноситься завжди кудись біжить, в цей самий час акселерометр має багато високо-частотних шумів (рис. 4.1.). Для оптимізації даних пристрою можна використати фільтри. Одним з фільтрів яким користуються часто це фільтр Калмана.

#### 4.2. Фільтр Калмана

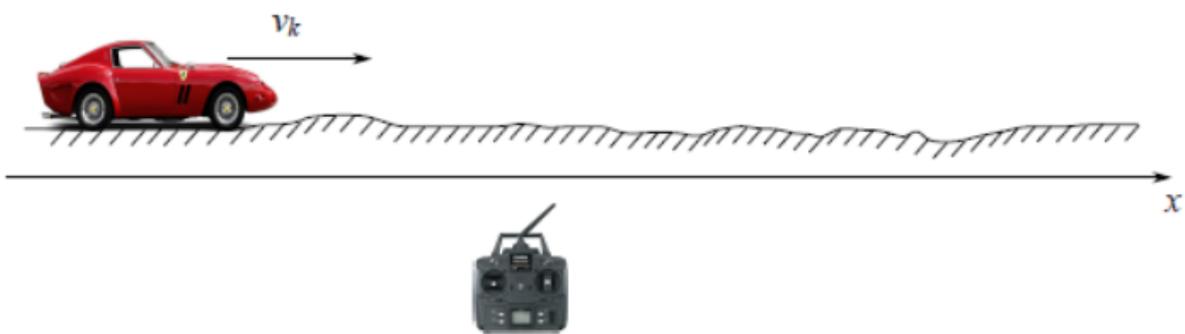
Фільтр Калмана – це алгоритмічний фільтр оптимізації даних пристрою, який дозволяє за допомогою обрахунків прибирати шуми та згладжувати графіки. Основний принцип роботи фільтра полягає в використанні ним інформації про фізичне походження даних. Для прикладу, при фільтрації даних про рух автомобіля, його інерційність дозволяє вважати різкий скачок його швидкості помилковим оскільки він не буде визначати саму постійну швидкість а лиш стрибок в певному проміжку часу. Фільтр Калмана дозволяє виокремити цей скачок і надати достовірнішу інформацію про швидкість на основі реальної швидкості автомобіля. Завдяки досить великій простоті використання фільтру він є можливо навіть і найкращим серед інших фільтрів даних. Також він є надзвичайно популярним в колах любителів електроніки завдяки тому, що існує безліч його варіацій, в тому числі – ще більш спрощені.



**Рисунок 4.1.** Дані акселерометра MPU6050

На картинці вище можна спостерігати шуми показників акселерометру. Зеленим кольором виділений показник по осі  $Z$  (вісь направлена вгору). Можна побачити велику кількість шумів, особливо їх кількість висока при русі гіроскопа. Завдяки фільтру Калмана ці шуми можна прибрати і отримати плавну криву положення сенсору.

Для покращення уявлення про те як працює алгоритм, скористаємося простим прикладом що часто зустрічається в мережі інтернет [5, 6].



**Рисунок. 4.2.** Схематичне зображення прикладу для пояснення алгоритму роботи фільтра Калмана

На рисунку вище, можна побачити зображення до прикладу. Припустимо що в нас є автомобіль який рухається вперед або назад. Будемо вважати що координати автомобіля рівні  $x_k$ . При цьому, ми знаємо масу автомобіля, напрям

його руху, форму, дорожнє покриття і т.д. Також ми знаємо як впливає на швидкість пульт керування цим автомобілем. Будемо вважати що це  $u_k$ . Таким чином, ми можемо знайти як буде змінюватися координата автомобіля. Вона буде відповідати закону  $x_{k+1} = x_k + u_k dt$ . Але це буде лише в ідеальних умовах де немає сторонніх сил. У випадку реального тестування в середовищі можна виявити що на рух автомобіля впливає багато показників які ми не враховували. Для прикладу це може бути вітер, палички та каміння на дорозі. Назвемо ці сили  $\varepsilon_k$ . Таким чином, формула буде мати наступний вигляд:

$x_{k+1} = x_k + u_k dt + \varepsilon_k$ . Крім цього, в автомобілі встановлений GPS який вимірює його положення але він робить це з похибкою  $n_k$ . Таким чином, формула для знаходження координат автомобіля за GPS буде наступна:

$$x_{k+1} = x_k + n_k.$$

Таким чином, ми маємо 2 формули для знаходження координати нашого автомобіля. Тепер нашим завданням є, знаючи неправильні показники  $z_k$  знайти максимально приближене до реальних показників  $x_k$ . Позначимо його як  $x_k^{opt}$ . Зазначу що показники можуть бути іншими для іншого випадку. Наші кінцеві формули виглядають так:

$$x_{k+1} = x_k + u_k + \varepsilon_k$$

$$z_k = x_k + n_k$$

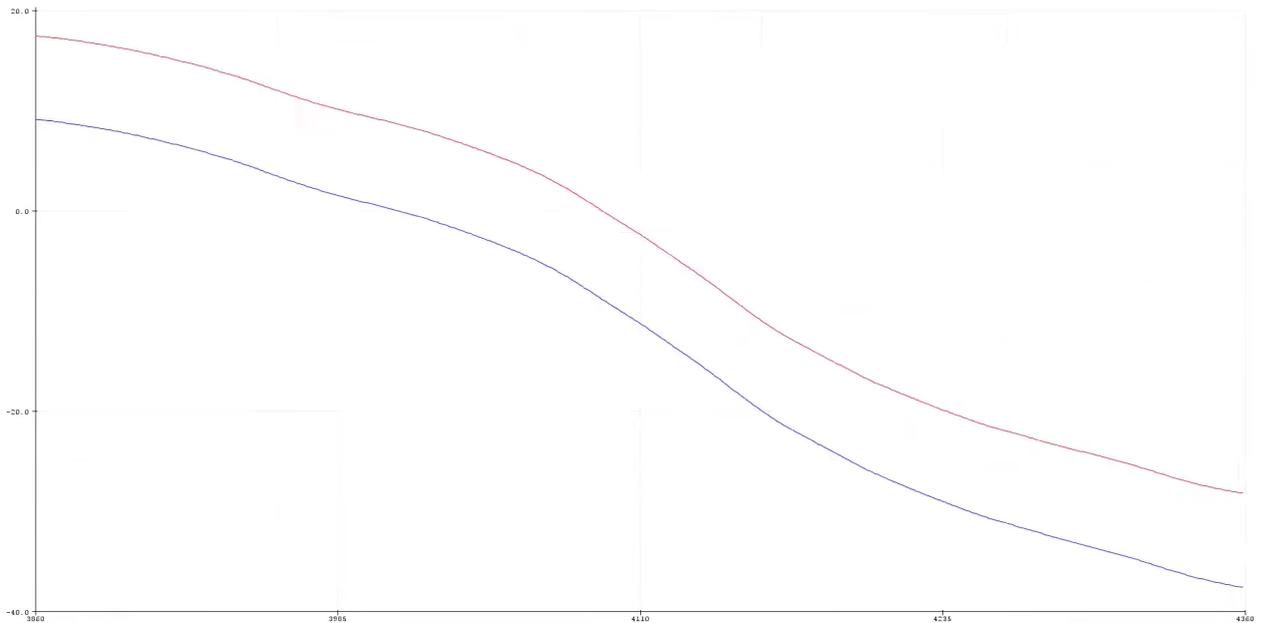
Уявимо собі що на  $k$  – кроці ми вже отримали коректні відфільтровані дані  $x_k^{opt}$  які добре передають правдиву координату  $x_k$  автомобіля. При цьому, ми не знаємо рівняння яке контролювало б зміну на невідомій координаті. Через це, не маючи інформації про дані сенсору, ми можемо передбачити показники на  $k + 1$  – кроці як дуже близькі до попередніх. Все ж. це не дозволить нам отримати більш точну інформацію. З іншого боку, на кроці  $k + 1$  ми матимемо дані з сенсора GPS,  $z_{k+1}$ . Ідея Калмана полягає в тому, щоб на основі найбільш наближеної координати до істинної,  $x_{k+1}$ , ми маємо обрати золоту середину між

$z_{k+1}$  (даними GPS на наступному кроці) і  $x_k^{opt} + u_k$  (наші прогнози стосовно даних GPS на наступному кроці). Зазвичай, для точного калібрування використовують коефіцієнти. Дано коефіцієнт  $K$  показникам сенсора а передбаченню  $1 - K$ . В результаті наша формула матиме наступний вигляд:  $x_{k+1}^{opt} = K * z_{k+1} + (1 - K) * (x_k^{opt} + u_k)$ . В цьому рівнянні  $K$  – називають коефіцієнтом Калмана. Варто зауважити що разом з коефіцієнтом необхідно писати його ітерацію ( $K_{k+1}$ ), проте ми цього робити не будемо щоб не ускладнювати вигляд формули але будемо мати її на увазі.

Коефіцієнт Калмана підбирається таким чином, щоб отримати максимально достовірні дані показників. Якщо ми для прикладу довіряємо більше сенсору ніж передбаченню то коефіцієнт сенсору буде близьким до 1 а передбачення – меншим або не сильно вищим за 0.5. Ці компоненти підбираються методом тестування. Проте в ідеальних умовах, якщо є можливість мінімізувати похибку ( $e_{k+1} = x_{k+1} - x_{k+1}^{opt}$ ) то він буде знаходитися як  $e_{k+1} = (1 - K)(e_k + \varepsilon_k) - K * n_{k+1}$ .

Хтось може запитати: “Що означає мінімізувати похибку? Ми ж яскраво та чітко бачимо що це випадкова величина яка щоразу приймає різні значення.”. Насправді це дійсно так, не існує єдиного підходу для пошуку того самого значення “мінімізації похибки”. Це теж саме що і з дисперсією випадкової величини при оцінках її розходження. Найпростіше використати середнє значення від квадрату похибки  $E(e_{k+1}^2) = (1 - K)^2(E * e_k^2 + o_\varepsilon^2) + K^2 * o_n^2$ . Таким чином, результуюча формула пошуку за фільтром Калмана буде мати наступний вигляд.

$$E(e_{k+1}^2) = \frac{o_n^2(E * e_k^2 + o_\varepsilon^2)}{E * e_k^2 + o_\varepsilon^2 + o_n^2} [7]$$



**Рисунок 4.3.** Дані опрацьовані фільтром Калмана. Відкалібрований (червоний), не відкалібрований (блакитна)

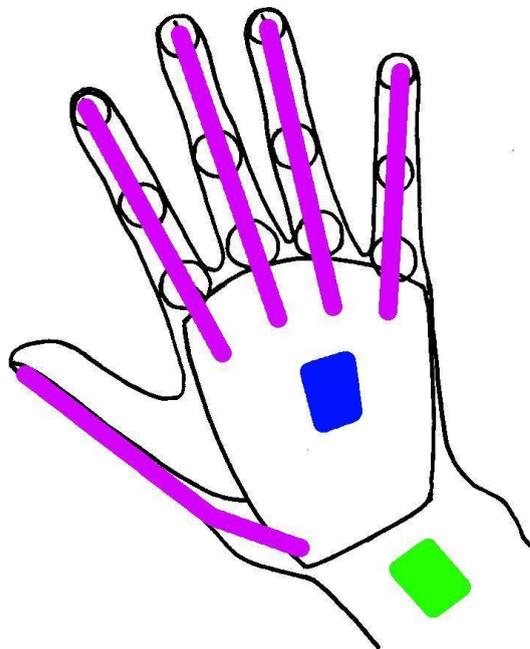
Використовуючи цей алгоритм але на даних з MPU з урахуванням даних акселерометру та гіроскопу, можна досягти надзвичайно великої точності показників (**рис. 4.3.**). Цієї точності та пряmolінійності буде цілком достатньо для наших потреб, а саме – тестування системи зчитування рухів руки.

### 4.3. Проектування системи

Під час проектування системи важливо було враховувати деякі особливості комплектуючих. В першу чергу це стосувалося тензорезисторів. Оскільки тензорезистори в своїй основі мають трубку, внутрішній діаметр якої близько 3.2 мм (трубка крапельниці) і стінки якої не товстіші за 1мм, то це спричинило проблему постійного заломлення трубки. Це перекривало світловий потік і зменшувало показники фоторезистора. Така проблема стосується багатьох варіантів трубок які в основі використовують подібні трубки. Рішення проти переломлення трубок є. Воно полягає в створенні рухомої основи яка дасть трубкам рухатися вільно в спеціально підготовлених пазах (трубках більшого діаметру намертво закріплених на рукавиці). Це дозволило зменшити ризик заломлення трубки. Навпроти – це спричинило помилку не стабільного руху трубок в середині пазів. Іншим рішенням даної

проблеми є – використання пружин в які поміщуватиметься трубка. Розміщуватися ці пружини мають в місцях де передбачається згин пальців. Вони мали б пом'якшувати кут згину трубки. Ще, серед способів вирішення цієї проблеми, може бути використання трубок з товстого матеріалу або менш гнучкого. В такому випадку потрібно мати баланс з силою згину і кутом згину. Якщо матеріал буде занадто грубим і твердим це спричинить помилку недогину трубки.

Взагалі, є багато проблем пов'язаних з використанням такого виду тензорезисторів. З багато з них було виявлено під час практичних тестів різних варіантів таких систем. Як одне з рішень, було розглянуто саморобний варіант фольгованого резистора. Більш детально про нього можна прочитати в розділі тензорезистори. Він показав погану стабільність але дозволив би створити досить компакту систему. В той самий час, компактність була б перевагою оскільки рукавиця має бути зручна та проста в використанні.



**Рисунок 4.4.** Схема розміщення комплектуючих

Іншою проблемою було розміщення комплектуючих на рукавиці. В особливості це стосувалося розміщення тензорезисторів на пальцях. Розглядалося 3 способи розміщення.

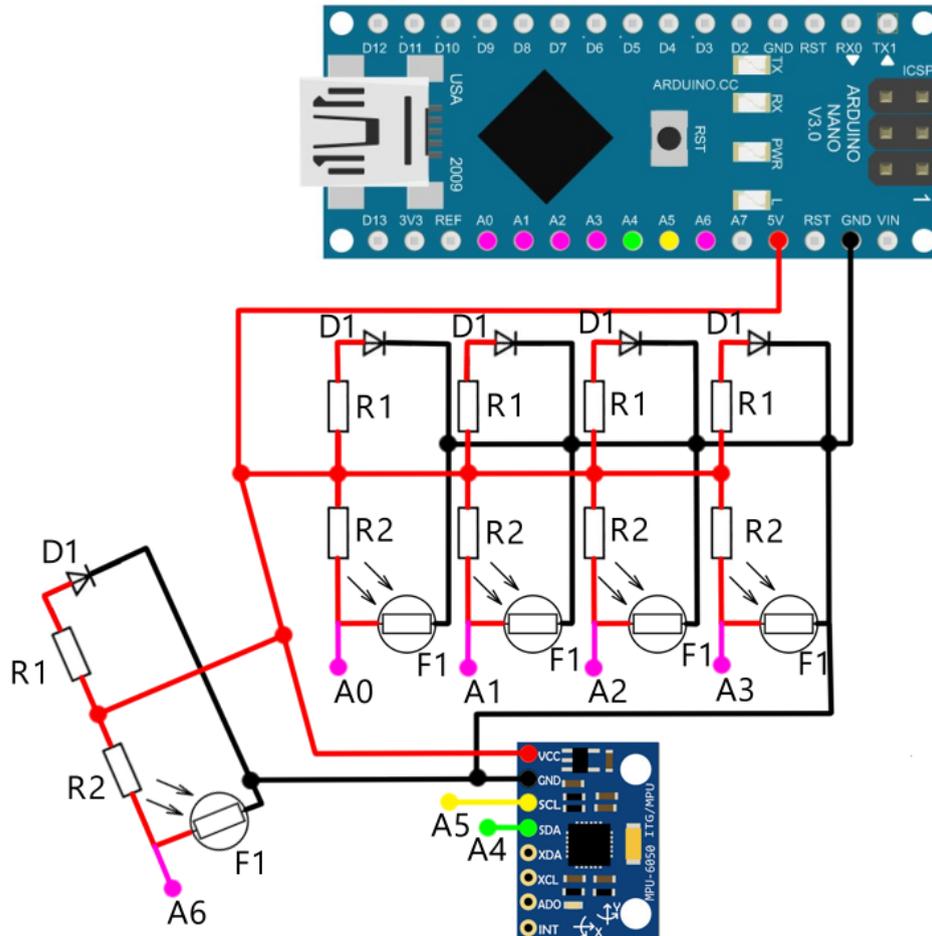
Один зі способів полягав в закріпленні не рухомої частини трубки на кістяшках пальців. Таке розміщення дозволило б знаходити згин всього пальця. Недоліком такого способу є зменшена точність рухів та незручність носіння рукавиці. Незручність мала вигляд банального прокручування трубок та погіршила б програмування. Оскільки при такому варіанті всі трубки були б досить довгими це спричиняло проблему прокручування іншого кінця трубки. Таким чином погіршувалась чутливість та згин трубок. Від цього способу було вирішено відмовитися.

Другим способом розміщення було закріплення трубок на кінцях пальців і відносна фіксація інших кінців на кістяшках. Це дозволило б зменшити проблему прокрутки трубок, але не прибрато б її повністю. Також при такому положенні частина що вільно рухається, часто впивалась в рукавиці тим самим зупиняючи рух і спричиняючи проблему некоректності даних. Перевагою такого способу є простота монтажу і зручне розміщення провідників.

Третій, як на мене найкращий, - закріплення трубки на першій фаланзі пальця з її вкороченням і виводом рухомих кінців на кінчик пальця. Це саму оптимальне розміщення "тензо-трубок" на пальці. Таким чином, можна оминати проблему пов'язану з прокручуванням трубок та заломленням. Надзвичайно вагомим недоліком є неможливість без зайвих пристроїв зчитати згин першої фаланги пальців. Цим було пожертвувано задля підвищення надійності і полегшення проектування. В подальшому цю проблему можна усунути за допомогою використання більшої кількості MPU6050 але на менших платах. Варто прибрати зайві компоненти та залишити лише ті, які необхідні для роботи сенсора. Таким чином вдасться зменшити розмір системи, що спростило б конструювання.

Крім цього, було розглянуто розміщення світлодіодів та фоторезисторів. Найкращим розміщенням фоторезисторів (було виявлено під час тестувань) є – розміщення в нерухомій частині конструкції. Це обумовлено їх проблемністю монтажу та необхідністю, стабільного розміщення в області трубки. Іншою мовою – їх фіксацією відносно трубки. Окрім цього, в даному варіанті поблизу

них розміщуються резистори, що також вимагає їх фіксації в нерухомому стані. Ідеальним розміщенням комплектуючих є схема на **рисунку 4.4**. Фіолетовим кольором зображені тензорезистори, блакитним MPU а зеленим I2S розширювач портів (TCA). В такому варіанті схеми не має потреби використовувати TCA. Розширювач потрібний лише для того, щоб в подальшому полегшити роботу системи.



**Рисунок 4.5.** Спрощена схема підключення комплектуючих в єдину систему, без врахування розширювача портів

Ознайомившись з тонкощами проектування, варто ознайомитися зі схемою підключення комплектуючих. На рисунку вище, можна побачити те як підключені комплектуючі в схемі. Принцип підключення був взятий з [8]. Для кращого розуміння їх номіналів, назви однакових за характеристиками компонент схеми, однакові. Якщо розглянути розміщення всіх комплектуючих та їх підключення до плати Arduino Nano, Стає зрозумілим порядок обробки

інформації. В першу чергу, кількість контактів необхідних для роботи системи становить 9. Більшість з них це аналогові входи з яких зчитується інформація в порядку: вказівний, середній, безіменний, мізинець і лиш тоді – великий палець. Такий порядок є зручним для розуміння всієї схеми та того, як відбувається обхід по аналоговим портам плати. MPU6050 опрацьовується окремо. Принципової різниці в порядку зчитування MPU – аналогові, аналогові – MPU немає. Важливим лиш є порядок її відправки, про це будемо говорити окремо в одному з розділів. Живлення системи відбувається з плати Arduino. Вона видає 5В. Ми використовуємо світлодіод 3.2мм з напругою живлення близько 3.1 – 3.3 вольт. Для того щоб понизити напругу на світлодіод було використано резистори номіналом 220 Ом (R1 на схемі). Такого опору більш ніж достатньо для роботи світлодіода. Фоторезистор. Більшість з нас знає що ця комплектуюча використовується як частина резистивного подільника (в більшості випадків). Резистивний подільник це два резистори (або комплектуючі створюють опір) з'єднані послідовно один за одним. До контактів кінців резисторів підведено напругу. Покази з подільника знімаються на з'єднаній між собою частині резисторів і мінусом живлення. З фоторезистором схема така ж сама. Відмінністю є лише полярність напруги. Якщо подавати напругу з боку фоторезистора то при збільшенні його опору напруга буде зменшуватися. Якщо напругу подавати з боку резистора, то при збільшенні опору фоторезистора напруга буде збільшуватися. Це варто враховувати. Номінал суміжного резистора з фоторезистором становить 10 кіло Ом (R2 на схемі).

Зі схем наведених вище, можна зробити висновок про простоту роботи пристрою (рис. 4.4., 4.5).

#### 4.4. Реалізація проекту

І так, після ознайомлення з теоретичною інформацією, можна переходити до реалізації проекту. Реалізацію можна поділити на 4 розділи: збирання тестового прототипу, програмування прошивки мікроконтролера, програмування драйвера і тестування.

#### 4.4.1. Прототипування

Прототипування приладу – це дуже важливий та цікавий процес. Метою цього процесу є просте збирання рукавиці для тестування роботи комплектуючих в системі. Важливо розуміти що прототипування це готове для продажу рішення. В даному випадку вимагається прототип який дозволить перевірити функціональність роботи системи і перевірити її програмування.

В системі передбачається велика кількість провідників. Всі вони мають мати досить мале січення кабелю. Підібрати подібне січення це складний процес. Більшість провідників має січення близько 1 – 1.5 мм. При цьому, 1 мм мають зазвичай дроти які використовують в маленьких трансформаторах. Їх проблема полягає в одножильності що спричинить багато проблем паяння та міцності (використання таких провідників погано себе зарекомендувало). Більш підходящим для такої системи буде багатожильний провід. Їх особливість полягає в складності знайти провід малого січення. Також важливо враховувати його гнучкість. Найгнучкішим варіантом багатожильних проводів є – проводи з силіконовою оплеткою. Для прототипу було обрано, тимчасово, провід одножильний товщиною 1 мм.

Після вибору основного провідника, варто зробити вибір головної комплектуючої, а саме – рукавиці. Правильний підбір рукавиці для тестування теж важливий. Хоч в релізного варіанті пристрою буде матеріал підбиратися з урахуванням необхідних властивостей, для тестового варіанту краще обрати готове рішення. Для таких цілей ідеально підходять будівельні варіанти рукавиць. Найкращим, з моєї точки зору, буде еластична рукавиця з матеріалу що буде добре прилягати до руки. Фото такої рукавиці можна побачити нижче.



**Рисунок 4.6.** Пристрій на основі еластичної будівельної рукавиці

Після підбору складових варто перейти до розділу паяння. Процес паяння тонких одножильних проводів досить складний. Зазвичай для такого роду робіт використовують паяльні кислоти по типу ортофосфатної кислоти. Паяння з використанням каніфолі не зовсім є підходящим. Також аспектом який варто враховувати є розміщення провідників та їх маркування для спрощення збирання системи. Збирання пристрою це довгий процес через велику кількість паяння та експериментів. Тут необхідно мати багато терпіння та велику усидливість.

#### 4.4.2. Програмування мікроконтролера

Від вибору мікроконтролера залежить те, в якому середовищі буде проводитися програмування. Якщо основою пристрою є STM32 то найбільш підходящим буде використання їх офіційних IDE. Серед них є: STM32CubeMX (початкове налаштування плати), STM32CubeIDE (середовище програмування). Окремо необхідно виділити Keil uVision5. Keil є дуже потужним інструментом який нічим не поступається своєму офіційному конкуренту. Також серед

любителів воно є найпопулярнішим. Окремо звернемо увагу на те, що є 2 типи програмування на STM32. HAL та LL. Кожна має свою перевагу але найлегше використовувати HAL.

Ми будемо використовувати Arduino nano. Arduino це найпопулярніший вид мікроконтролерів серед любителів. В масових розробках його замінюють здебільшого на STM або на процесори з власною архітектурою. Також для ATmega328P є безкінечна кількість рішень та бібліотек майже для всього. Багато хто радить починати з Arduino але це не так. Для кращого розуміння роботи комплектуючих краще все ж починати з STM або PIC. Це займе набагато більше часу але дозволить розуміти принципи програмування та краще ознайомитися з комплектуючими які використовуються.

Розглянемо уривок коду:

```
void TCA9548A(uint8_t bus){
  Wire.beginTransmission(0x70);
  Wire.write(1 << bus);
  Wire.endTransmission();
}
```

Ця частина пояснює роботу мультиплексора TCA9548A. Його використання в схемі не обов'язкове але дозволяє значно полегшити подальше покращення рукавиці. MPU6050\_light.h – це бібліотека для роботи з MPU. Якраз наявність таких бібліотек це і є перевага Arduino перед іншими мікроконтролерами.

В цій бібліотеці є всі необхідні функції для роботи з MPU. Тут наявні функції зчитування інформації та обрахунку її за допомогою фільтру Калмана. В розділі “фільтр Калмана” можна ознайомитися з його ідеєю використання передбачення в знаходженні коректних даних. Дуже раджу ознайомитися з цим розділом і переглянути приклад який представлений в ньому.

Робота з аналоговими виходами є дуже простою. Для того щоб зчитати дані з конкретного виходу, варто лиш викликати функцію `analogRead` (`analog_ip1`). Вона поверне рівень напруги на вході з її конвертацією від 0 до 1032. Це теж саме що і 0 – 5 вольт.

В зв'язку з обміном інформацією через USB, потрібно додати те, що налаштування послідовного порту виходу проводиться в головному циклі `setup()`. При налаштуванні вказується швидкість обміну інформацією. Цю швидкість необхідно враховувати в програмуванні драйвера COM Reader. Саме те чи співпадає швидкість буде означати наскільки коректно приходять дані пристрою. Особливістю роботи з нашим драйвером також є те що ми використовуємо символ закінчення рядка “.”. Це необхідно щоб програма мала уявлення про те чи закінчили ми передачу стрічки інформації через послідовний вихід. Більш детально про роботу драйвера можна прочитати в наступному розділі.

#### **4.4.3. Програмування драйвера**

Що мається на увазі під словом драйвер. Драйвер це система зчитування і конвертації сухих даних приладу в дані які зрозумілі для обробки програмою, комп'ютером та периферією. Фактично можна сказати що це алгоритм конвертації незрозумілих даних в зрозумілі. Драйвер написаний на мові C++.

Для багатьох програмістів мова C++ викликає дуже погані спогади. Вони часто намагаються уникати цю мову і переходять до простіших мов програмування. Попри це, C++ є дуже швидкою мовою та популярною серед програм на Windows і Linux. Дуже складним для початківців є те, що ця мова вважає користувача професіоналом. Більшість компіляторів не пояснюють помилки зрозумілою мовою, навпаки – надають код помилки та загальний опис який, як показує практика, не підходить саме для цього випадку. Тому, поріг входження в програмування на мовах C високий. Там унікальний синтаксис і правопис який варто знати. Сильно виділяється архітектура програм на C. Вона є об'єктно орієнтованою. Сама мова підводить користувача до того, що він зобов'язаний використовувати класи та структури для полегшення програмування та структурування програм.

Com Reader – це сервіс для роботи з даними які надходять з послідовного порту. В мовах C, робота з COM надзвичайно схожа до роботи з файлами [9]. Наведемо частинку коду драйвера.

```
connect_port = CreateFile(
    ("\\\\.\\\\" + com_port).c_str(),
    GENERIC_READ | GENERIC_WRITE,
    0,
    NULL,
    OPEN_EXISTING,
    FILE_ATTRIBUTE_NORMAL,
    NULL
);
```

Вищенаведений код створює файл для читання інформації з необхідного порту. Змінна `com_port` це назва порту. Вона задається як COM + номер послідовного порту. В персональних комп'ютерах наявно близько 255 портів. Для коректної роботи з портом, необхідно провести налаштування параметрів взаємодії з ним. При роботі з файлами така потреба виникає рідко, а при роботі з ком портом це є обов'язковим.

```
DCB serial_params;
serial_params.DCBlength = sizeof(serial_params);
serial_params.BaudRate = baud_rate;
serial_params.ByteSize = 8;
serial_params.StopBits = ONESTOPBIT;
serial_params.Parity = NOPARITY;
```

Вище наведено код налаштувань параметрів швидкості (`baud_rate`), кількість бітів (8) кількість бітів завершення і так далі. Головним тут є `baud_rate`. Він вказує на швидкість обміну інформацією через ком порт. Цей показник має співпадати з показником прошивки. Ми його вказували при налаштуванні ком порту.

```
COMMTIMEOUTS set_timeout;
set_timeout.ReadIntervalTimeout = 1;
set_timeout.ReadTotalTimeoutConstant = 1;
set_timeout.ReadTotalTimeoutMultiplier = 1;
set_timeout.WriteTotalTimeoutConstant = 1;
set_timeout.WriteTotalTimeoutMultiplier = 1;
```

Вищенаведений код є кодом налаштування очікувань. Зазвичай його встановлюють як 1 але він залежить від самого проекту тому і підбирається під нього окремо.

```

DWORD byte_iterate;

char resived;
std::string s;
while (ReadFile(connect_port, &resived, 1, &byte_iterate, 0))
{
    if (byte_iterate > 0 && resived != ';') {
        s += resived;
    }
    else {
        return s;
    }
}

```

Код наведений вище це код зчитування даних з ком порту. Як можна помітити головним показником закінчення передачі даних є знак “;”. Він позначає кінець стрічки. Саме читання файлу відбувається таким само чином як і звичайного txt документу. Єдине що важливо це те, що налаштування параметрів опрацювання інформації залежить напряду від побудови алгоритму спілкування пристроїв. Додатково для покращення роботи можна додати сюди час життя запиту. Це буде щось схожу на TTL post або get запиту. Він дозволить виходити з циклу раніше ніж прийде інформація. Це збільшить кількість кадрів програми. В нашому випадку такої критичної потреби немає тому такий код нам цілком підійде.

Однією з корисних функцій для роботи буде функція пошуку доступних портів. Вона дозволить знайти доступні порти і обрати той який нам буде найцікавіший. Також за її допомогою можна побудувати автозахоплення пристрою. Це може виглядати як прохід по існуючих портах, кидання запиту туди для встановлення зв'язку та зчитування відповіді. Якщо відповідь є і вона така ж як нам потрібно то ми захоплюємо пристрій. Нище наведено код який для цього підійде якнайкраще. Тут ми знаходимо назву порту і записуємо її в вектор для подальшого відображення в IMGUI UI.

```

std::vector<std::string> arr;

char lpTargetPath[5000];

for (int i = 0; i < 255; i++)
{
    std::string str = "COM" + std::to_string(i);
    DWORD test = QueryDosDevice(str.c_str(), lpTargetPath, 5000);

    if (test != 0)
    {
        arr.push_back(str);
    }

    if (::GetLastError() == ERROR_INSUFFICIENT_BUFFER)
    {
    }
}
return arr;

```

Також важливо пояснити те як працює інтерфейс обробки інформації. Важливим в цьому питанні є те, що обробка має працювати в іншому потоці від основного. Це дозволить стабілізувати кількість кадрів програми і отримати максимально плавну картинку. В нашому випадку немає необхідності для роботи в 200+ кадрів тому нам буде достатньо 10-15 кадрів. Для зображення роботи системи цього буде досить. Код прикріплений нижче.

```

static std::mutex m;

m.lock();
std::string s;
s = COM::SerialRead();
com_info = s;

m.unlock();

```

Алгоритм зчитування інформації по черзі а не паралельно. Він необхідний щоб не виникало випадків коли не актуальні дані приходять після актуальних. Для цього ми блокуємо обробку нових до поки не прийшли дані з того запиту який ми кидали перед цим.

```

void com_Reader() {
    std::queue<std::future<void>> queue;

```

```

if (queue.empty()) {
    queue.push(std::async(std::launch::async, _com_reader));
}
else {
    queue.front().wait();
    queue.pop();
}
}

```

Використання `queue` тут нам буде досить корисним а `future` дозволяє обирати те як ми будемо працювати з потоками. Всього є декілька видів роботи з потоками але ми не будемо концентруватися на них оскільки для нас достатньо `std::launch::async`. Він створює процес паралельний основному потоці, можна сказати асинхронний процес, для обробки інформації. Таку функцію доцільно використовувати щоб у всіх доступних потоках оброблялась своя інформація. Якщо використовувати 1 потік то швидкість буде страждати бо процесор буде використовуватися не на повну. Є процесори які мають 16 потоків і більше і уявити що робота відбувається не у всіх 16 а в 1 – це буває боляче. Можна сказати що процесор працює на 1/16 від своєї потужності.

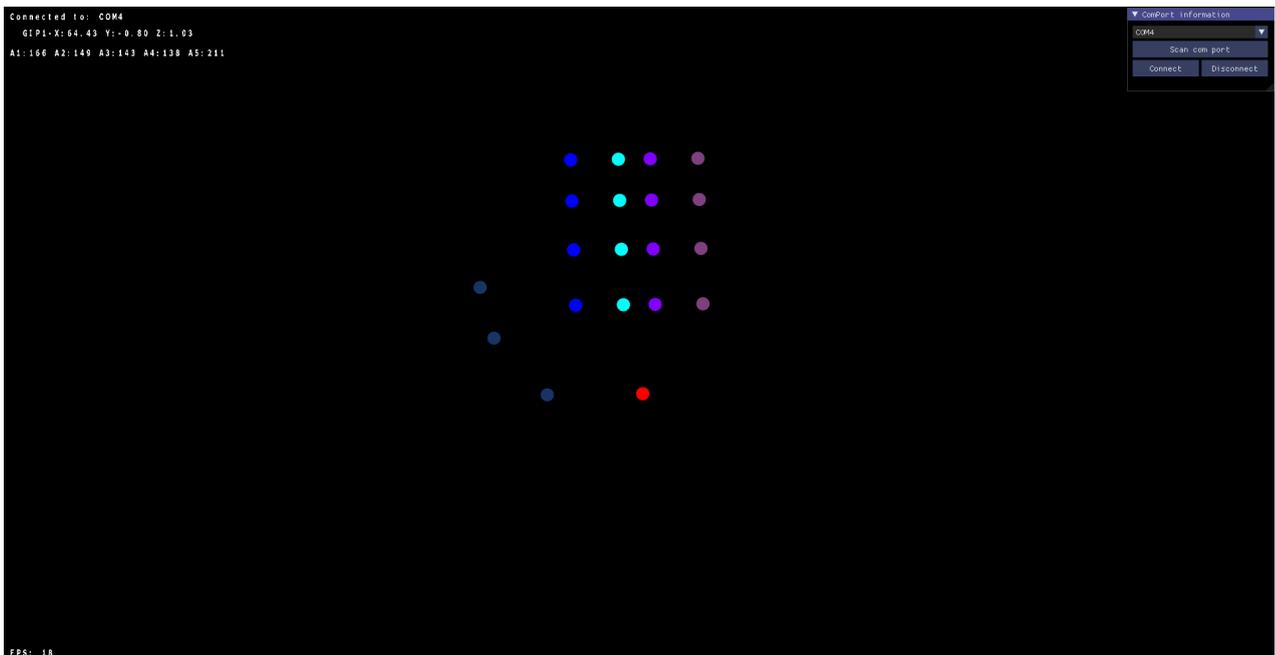
Для відображення всього інтерфейсу використовується технологія `ImGui` та `OpenGL`. `OpenGL` дозволяє працювати з 3D графікою а `ImGui` дозволяє створювати інтерфейс у необхідному для нас вікні.

## РОЗДІЛ V. РЕКОМЕНДАЦІЇ ЩОДО ВИКОРИСТАННЯ

### 5.1. Інструкція користувача

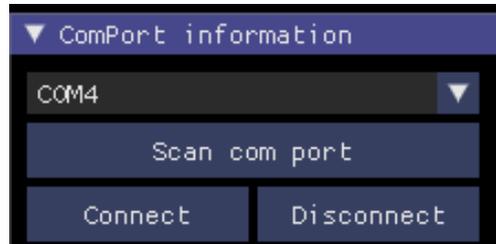
Пристрій дуже простий у використанні. Для того, щоб їм скористатися достатньо одягнути рукавицю, витягнути руку вперед для калібрування, підключити її до комп'ютера за допомогою USB і вибрати відповідний ком порт. Плата Arduino в такому випадку починає мигати що буде свідчити про те що калібрування пройшло успішно. Розглянемо більш детально.

Пристрій виглядає як звичайна рукавиця з трубками на пальцях (**рис. 4.6.**). Вона одягається так, щоб трубки знаходилися на тильній частині долоні. Після надягання варто визначити початковий напрям який програма буде вважати напрямком вперед. Для цього варто покласти руку на стіл перед собою. Далі необхідно підключити пристрій до usb порту і під'єднати до програми вибравши відповідний ком порт. Після 5 секунд калібрування система почне передавати інформацію до пристрою. Приклад інтерфейсу наведено на **рисунку 5.1.** Інтерфейс підключення знаходиться в спеціальному меню яке рухається по вікну області екрану, це дозволяє за потреби поставити його туди де воно буде зручно розміщене.



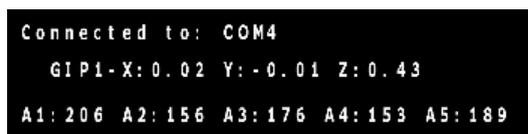
**Рисунок 5.1.** Приклад інтерфейсу програми

Вибір пристрою в програмному середовищі відбувається за допомогою Comport information (рис 5.2.). Тут є контекстне меню та кнопки сканування портів, підключення порту, відключення порту. Функція сканування доступних портів знаходиться в розділі “програмування драйвера”.



**Рисунок 5.2.** Інтерфейс обробки СОМ портів

Окремо варто виділити те, що зліва зверху можна побачити інформацію яка надходить з портів. Перший рядок показує те що приходить з MPU а другий – дані тензорезисторів. Це необхідно щоб покращити виявлення помилок пристрою. Ця опція є корисною і допомагає орієнтуватися в системі (рис. 5.3.).



**Рисунок 5.3.** Інформація яка надходить з пристрою

Після підключення пристрою, можна виконувати різного роду жести. Програма яка демонструє роботу не є ідеально точною тому це варто враховувати. Попри її недоліки, її достатньо для демонстрації роботи системи. Вона чудово візуалізує те, що пристрій працює. Теоретично її можна модифікувати для покращення демонстрації. Для цього необхідно додати можливість роботи з кістковою анімацією. Драйвер який використовується добре інтегрується в інші програми які пишуться на С подібній мові. Він простий і для того, щоб написати його навіть самостійно багато досвіду програмування не потрібно. Таку систему можна інтегрувати для прикладу в Unity чи unreal Engine.

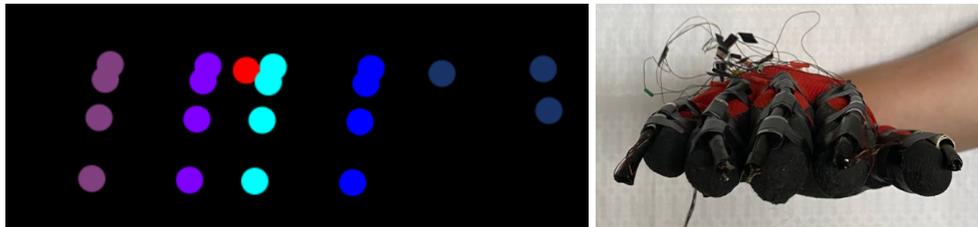
Особливістю рукавиці є можливість встановлення свого початкового положення вручну. Для цього необхідно повернути кисть на 90 градусів у бік, так щоб площина руки була перпендикулярна землі. Після чого, руку можна

рухати вправо або ліво. Таким чином можна змінити орієнтацію руки в програмі.

## 5.2. Тестування людино-машинної взаємодії

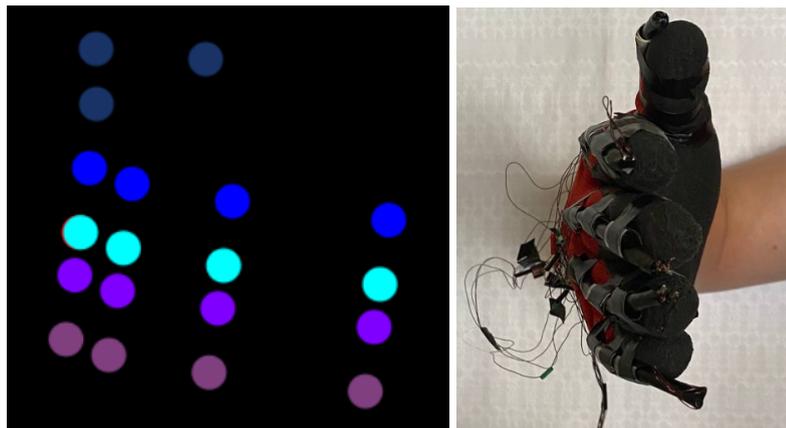
Тестування людино машинної взаємодії – це процес знаходження відповідностей і не відповідностей між роботою приладу та очікуваним результатом. Для проведення тестування варто розуміти те як правильно має реагувати програма на дії користувача та який результат є очікуваним. Для кращого розуміння, можна розглянути зображення.

Проведемо тестування зчитування положення руки в просторі. Для даного тесту нам необхідно встановити положення руки в початкове. Варто зазначити, що програма реагує дзеркально.



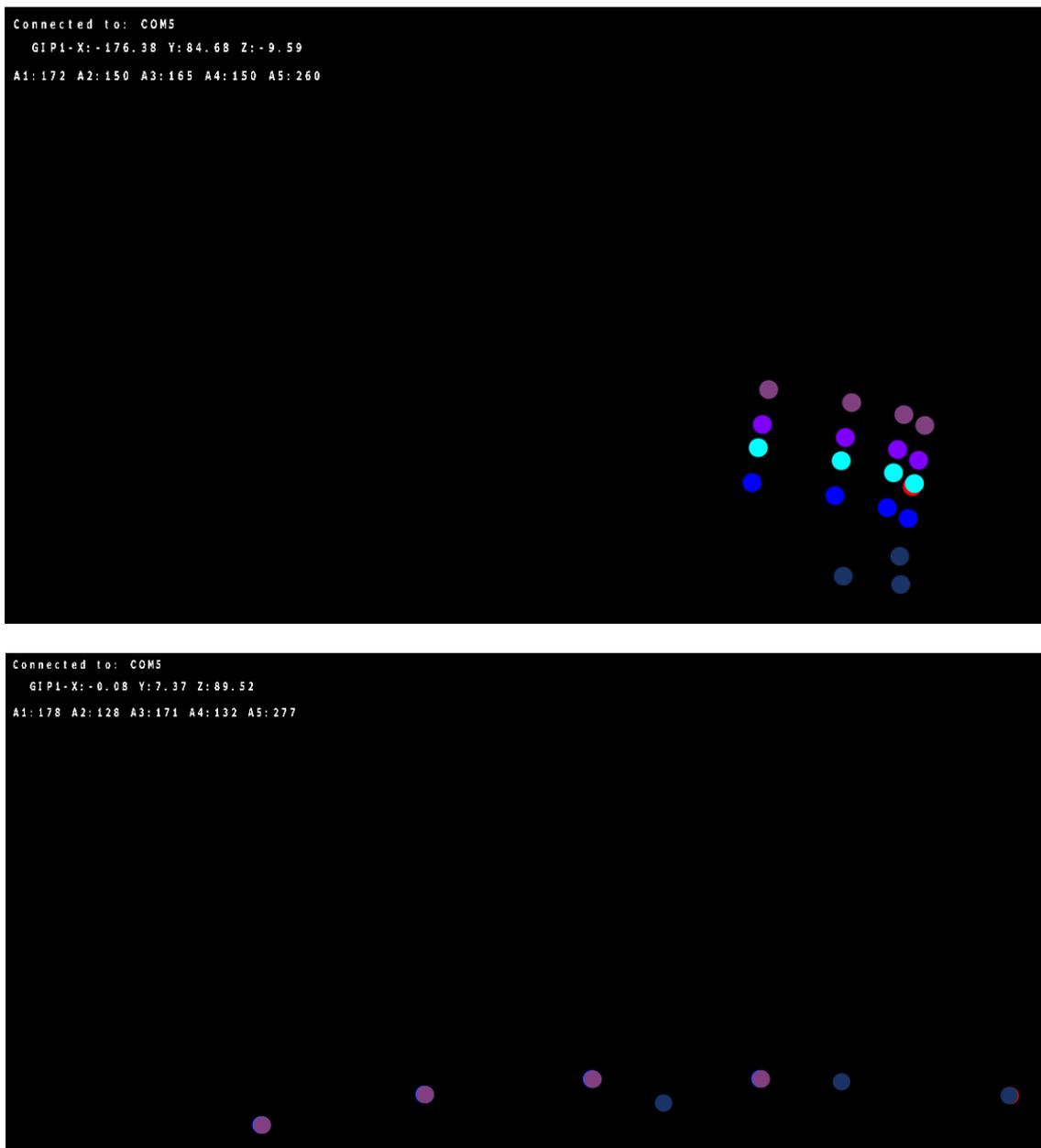
**Рисунок 5.4.** Початкове положення руки в програмі

Після того, як ми встановили положення руки в початкове, повернемо кисть на 85 градусів по годинниковій стрілці.

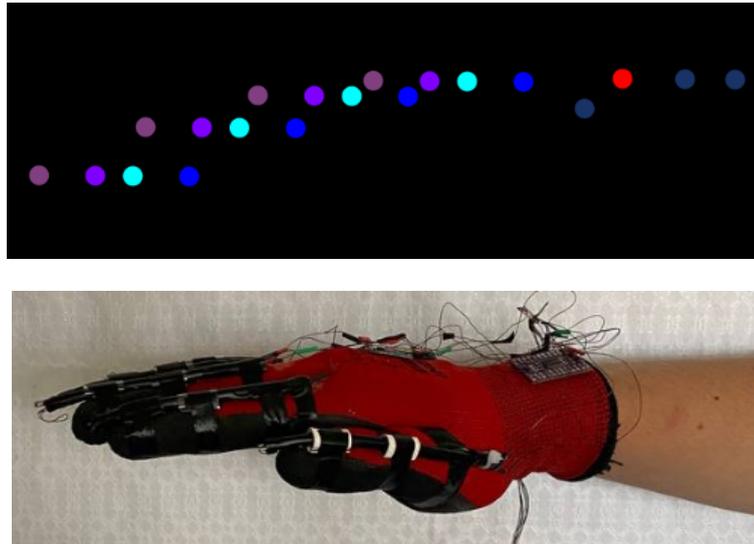


**Рисунок 5.5.** Положення руки при повороті кисті на 85 градусів

В нашому пристрої є налаштована система калібрування положення. Вона необхідна щоб ми могли відредагувати положення у випадку якщо є накопичені помилки. Для того щоб це зробити, необхідно повернути руку на 90 градусів, після чого, рука перевернеться і змінюючи напрям ми можемо його калібрувати.



**Рисунок 5.6.** Положення руки в 90 градусів та зміна показників по осі Z  
 Після того, як ми повернемо руку в початкове положення можна спостерігати зміну показників по осі Z. Верхній лівий кут згори. Наступним буде тест на поворот. Для даного тесту повернемо руку вправо на 45 градусів зберігаючи горизонт. Для кращого розуміння, звертаємо увагу на червону точку яка є центром руки.



**Рисунок 5.7.** Зображення руки після повороту за горизонтом на 45 градусів  
Як можна розуміти, система орієнтації в просторі працює коректно.

Проведемо тестування зчитування пальців руки. Для цього, в початковому положенні руки будемо згинати пальці по черзі в відповідній послідовності. Перш за все зігнемо вказівний палець.



**Рисунок 5.8.** Результат згину вказівного пальця

Далі зігнемо вказівний та безіменний разом.



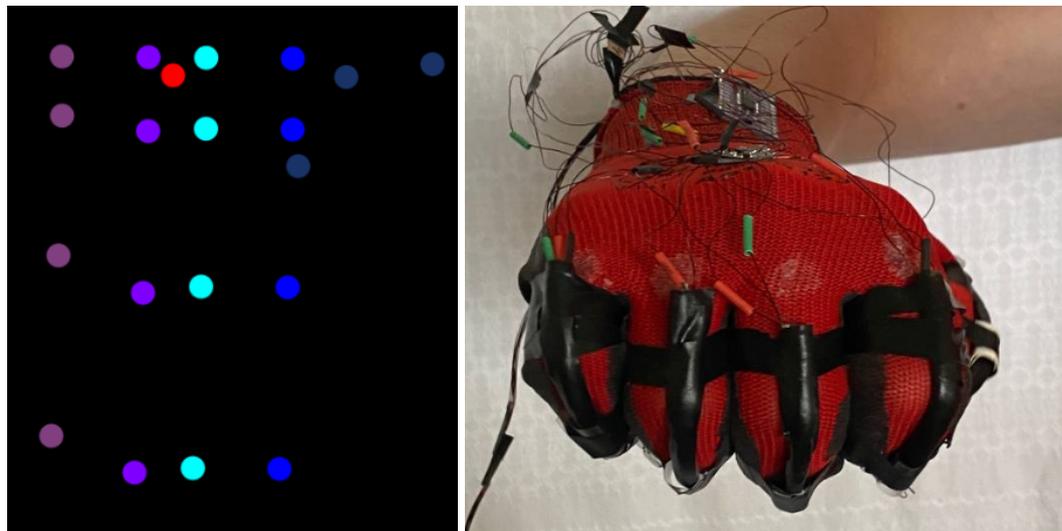
**Рисунок 5.9.** Результат згину вказівного та безіменного пальців

Тепер зігнемо великий палець і мізинець.



**Рисунок 5.10.** Результат згину великого пальця та мізинця

Тепер всі пальці разом.



**Рисунок 5.11.** Результат згину всіх пальців в кулак

Як можна спостерігати, є певні недосконалості програми що зображує згин пальців. Незважаючи на проблему, рукавиця працює справно та без критичних помилок. Таким чином можна говорити про дієвість такого способу реалізації приладу зчитування. Ну і для цікавості декілька прикладів комбінації положень пальців та руки.

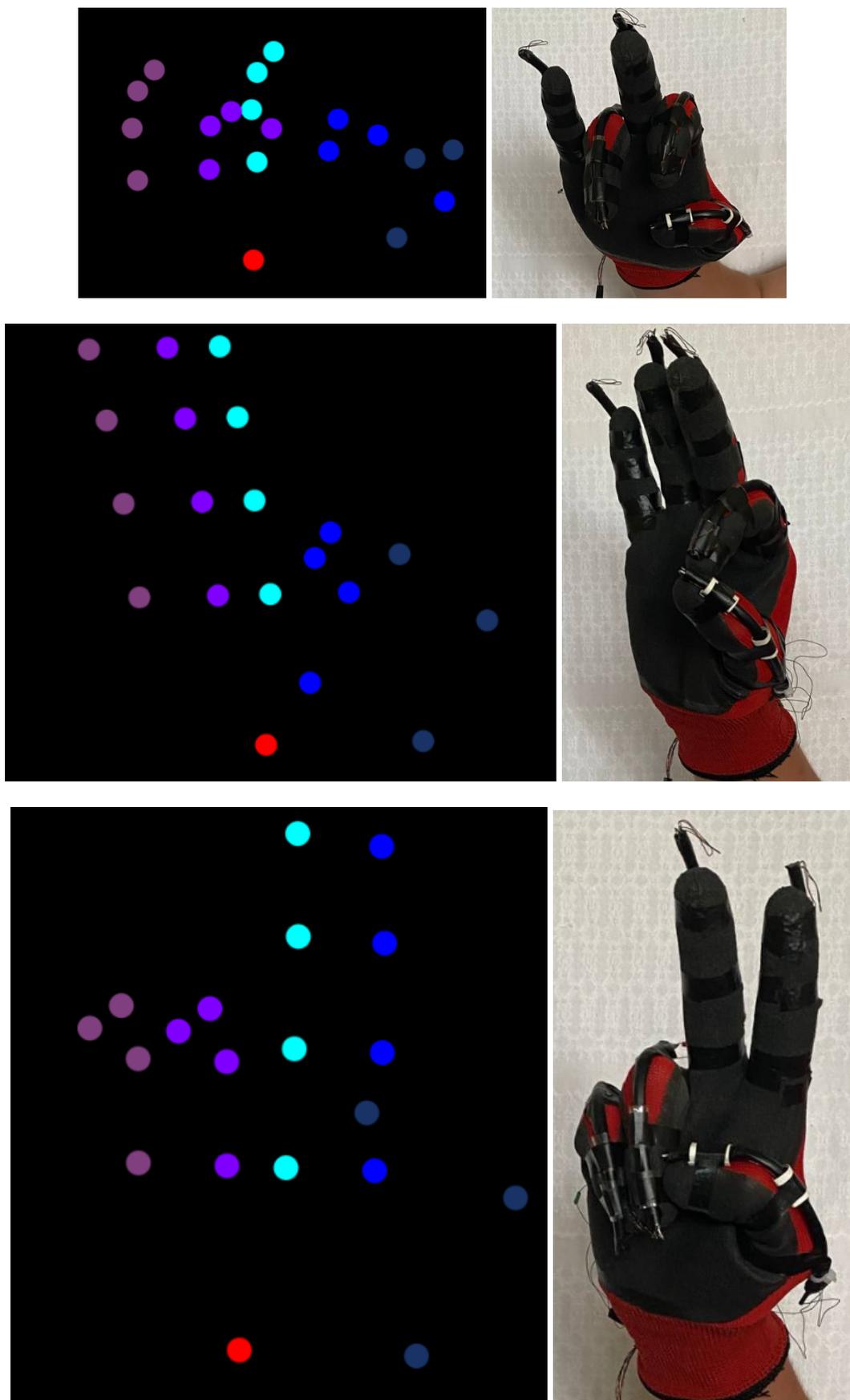


Рисунок 5.12. Приклади різних жестів руки

## ВИСНОВКИ

Робота, яка була проведена, дозволила отримати повноцінний пристрій для зчитування жестів руки. Було сконструйовано рукавицю, що повністю відповідає поставленим вимогам. Також, було запрограмовано мікроконтролер що отримує інформацію від сенсорів, обробляє її, та надсилає підготовлені пакети даних на драйвер операційної системи. В свою чергу, розроблений драйвер має функціональні можливості не тільки для роботи з даними пристрою, а й в цілому є універсальним в контексті роботи з СОМ портами. Він дозволяє коректно зчитувати інформацію з СОМ порта та передавати її в частини коду, що відповідають за подальшу обробку. Крім цього, в результаті нашого дослідження, було отримано відповідні практичні напрацювання, які можуть бути застосовані до нового, покращеного пристрою. Важливою перевагою такого приладу є простота та великий потенціал для покращення. Одним з покращень може бути використання більшої кількості сенсорів положення. Це дозволить збільшити точність системи.

Розроблений пристрій, при перенесенні комплектуючих на власні плати та додавання гнучких шлейфів, може стати досить конкурентоздатним на ринку. Він добре підійде для віртуальної реальності, в якості контролера і буде корисним при пілотуванні дронів чи управлінні роботизованими захватами. Таким чином, можна зробити висновок, що завдяки простій конструкції і малим габаритам, рукавиця заслуговує уваги потенційних інвесторів.

Варто розглянути можливі покращення системи. Для початку, можна замінити трубки на світловоди з власним, необхідним для системи, показником розсіювання. Також, варто розглянути додавання до системи більшої кількості сенсорів положення, це дозволить збільшити точність. Все це можливе завдяки правильно вибраному патерну проектування пристрою. Одна з вимог яка ставилася - це простота і широка можливість для покращень.

Під час роботи, було отримано масу важливих напрацювань. Одними з таких напрацювань, стали дослідження тензорезисторів. Було виявлено ряд особливостей фольгових та оптичних сенсорів. Для прикладу, фольговані

тензорезистори мають низьку точність і не можуть використовуватися в системах де необхідно отримувати показники в сотих одиниці. Це пов'язано з їх конструктивною особливістю. Також, це стосується і дротяних сенсорів. Вони не мають потенціалу для використання в системах де згинання відбувається більше ніж на 45 градусів. Це пов'язано з їх розмірами та, в більшості, з однонаправленістю роботи і початковим призначенням. Оптичні є одним з перспективних напрямків. Таке передбачення пояснено їх особливістю роботи. Вони дозволяють отримати досить точні дані і відобразити навіть малі зміни з високою точністю. Цей вид тензорезисторів має найбільший потенціал.

Підводячи підсумки, можна говорити що сконструйована система може бути корисною в мехатроніці, управлінні дронами, сфері розваг та кіно. Це дозволяє оцінити перспективи системи та демонструвати MVP реально зацікавленій аудиторії потенційних користувачів та інвесторів.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) J. V. Candy, Accelerometer Modeling in the State-Space. Lawrence Livermore National Laboratory. February 10, 2021.
- 2) Wikipedia [Електронний ресурс] : [Веб-сайт] – Гіроскоп – Електронні дані. Режим доступу : <https://uk.wikipedia.org/wiki/%D0%93%D1%96%D1%80%D0%BE%D1%81%D0%BA%D0%BE%D0%BF> (дата звернення 01.04.2023) – Назва з екрану.
- 3) Wikipedia [Електронний ресурс] : [Веб-сайт] – Лазерний гіроскоп – Електронні дані. Режим доступу: [https://uk.wikipedia.org/wiki/%D0%9B%D0%B0%D0%B7%D0%B5%D1%80%D0%BD%D0%B8%D0%B9\\_%D0%B3%D1%96%D1%80%D0%BE%D1%81%D0%BA%D0%BE%D0%BF](https://uk.wikipedia.org/wiki/%D0%9B%D0%B0%D0%B7%D0%B5%D1%80%D0%BD%D0%B8%D0%B9_%D0%B3%D1%96%D1%80%D0%BE%D1%81%D0%BA%D0%BE%D0%BF) (дата звернення 02.04.2023) - Назва з екрану.
- 4) Um.co.ua [Електронний ресурс] : [Веб-сайт] – Тензорезистори опору та вимірювальні схеми – Електронні дані. Режим доступу: <http://um.co.ua/8/8-2/8-22811.html> (дата звернення 05.04.2023) - Назва з екрану.
- 5) Wikipedia [Електронний ресурс] : [Веб-сайт] – Фільтр Калмана. – Електронні дані. Режим доступу: [https://uk.wikipedia.org/wiki/%D0%A4%D1%96%D0%BB%D1%8C%D1%82%D1%80\\_%D0%9A%D0%B0%D0%BB%D0%BC%D0%B0%D0%BD%D0%B0](https://uk.wikipedia.org/wiki/%D0%A4%D1%96%D0%BB%D1%8C%D1%82%D1%80_%D0%9A%D0%B0%D0%BB%D0%BC%D0%B0%D0%BD%D0%B0) (дата звернення 21.04.2023) - Назва з екрану.
- 6) В. І. Кожешкурт, В. В. Юзефович. Математичні методи обробки даних. Дослідження схем фільтрації алгоритмів трасової обробки інформації в системах моніторингу динамічних об'єктів: Інститут проблем реєстрації інформації НАН України, 25 November 2010. 10 ст.
- 7) R. E. Kalman. A new Approach to Linear Filtering and Prediction Problem: Research Institute for Advanced Study, 2 Baltimore, Md, 20 January 2002, 12 ст.

- 8) Jacques Frottit, Stefan Marks, Andy Connor. An Intuitive Tangible Game Controller: The 10th Australasian Conference on Interactive Entertainment (IE2014), 02 December 2014, 7ст.
- 9) Microsoft learn [Електронний ресурс] : [Веб-сайт] – Serial Communications – Електронні дані. Режим доступу: [https://learn.microsoft.com/en-us/previous-versions/ff802693\(v=msdn.10\)?redirectedfrom=MSDN](https://learn.microsoft.com/en-us/previous-versions/ff802693(v=msdn.10)?redirectedfrom=MSDN) (дата звернення 10.05.2023) - Назва з екрану.
- 10) Theverge [Електронний ресурс] : [Веб-сайт] – Meta’s sci-fi haptic glove prototype lets you feel VR objects using air pockets – Електронні дані. Режим доступу: <https://www.theverge.com/2021/11/16/22782860/meta-facebook-reality-labs-soft-robotics-haptic-glove-prototype> (дата звернення 01.04.2023) - Назва з екрану.

## ДОДАТКИ

Додаток 1  
Лістинг коду драйвера

### *Kod ComReader-a:*

```
#include "COM.h"
#include <windows.h>
#include <string>

#include "Printer.h"

HANDLE connect_port;

std::string COM::com_port = "COM1";
int COM::baud_rate = 9600;
char COM::buff[256];
bool COM::isCOM = false;
std::string COM::com_status = "";

int COM::SerialBegin(std::string _com_port, int _baud_rate) {
    com_port = _com_port;
    baud_rate = _baud_rate;

    com_status = com_port;

    if (com_port == _com_port && isCOM == true)
        CloseHandle(connect_port);

    connect_port = CreateFile(
        ("\\\\.\\\\" + com_port).c_str(),
        GENERIC_READ | GENERIC_WRITE,
        0,
        NULL,
        OPEN_EXISTING,
        FILE_ATTRIBUTE_NORMAL,
        NULL
    );

    if (connect_port == INVALID_HANDLE_VALUE) {
        return -4;
    };

    DCB serial_params;
    serial_params.DCBlength = sizeof(serial_params);
    if (!GetCommState(connect_port, &serial_params)) {
        return -3;
    }

    serial_params.BaudRate = baud_rate;
    serial_params.ByteSize = 8;
    serial_params.StopBits = ONESTOPBIT;
    serial_params.Parity = NOPARITY;

    if (!SetCommState(connect_port, &serial_params)) {
        return -2;
    }

    COMMTIMEOUTS set_timeout;
    set_timeout.ReadIntervalTimeout = 1;
    set_timeout.ReadTotalTimeoutConstant = 1;
    set_timeout.ReadTotalTimeoutMultiplier = 1;
    set_timeout.WriteTotalTimeoutConstant = 1;
}
```

```

set_timeout.WriteTotalTimeoutMultiplier = 1;

if (!SetCommState(connect_port, &serial_params)) {
    return -1;
}
return 0;
}
void COM::ConnectRead(std::string _com_port, int _baud_rate) {

    if (isCOM)
    {
        CloseHandle(connect_port);
        com_status += " disconnected";
        isCOM = false;
    }
    switch (SerialBegin(_com_port, _baud_rate))
    {
    case -4:
        com_status += " connecting error";
        break;
    case -3:
        com_status += " (1)get state error";
        break;
    case -2:
        com_status += " (2)get state error";
        break;
    case -1:
        com_status += " set timeout error";
        break;
    case 0:
        com_status = "Connected to: " + com_status;
        isCOM = true;
        return;
        break;
    }

    CloseHandle(connect_port);
}
std::string COM::SerialRead() {
    if (!isCOM) return "Com disconnected";
    if (!SetCommMask(connect_port, EV_RXCHAR)) { return ""; }

    DWORD byte_iterate;

    char resived;
    std::string s;
    while (ReadFile(connect_port, &resived, 1, &byte_iterate, 0))
    {
        if (byte_iterate > 0 && resived != '\0') {
            s += resived;
        }
        else {
            return s;
        }
    }

    return "";
}
int COM::Disconnected() {
    if (isCOM == true) {
        CloseHandle(connect_port);
        com_status = "";
        isCOM = false;
        return 1;
    }
    return 0;
}

```

```

}
std::vector<std::string> COM::FindComPort()
{
    std::vector<std::string> arr;

    char lpTargetPath[5000];

    for (int i = 0; i < 255; i++)
    {
        std::string str = "COM" + std::to_string(i);
        DWORD test = QueryDosDevice(str.c_str(), lpTargetPath, 5000);

        if (test != 0)
        {
            arr.push_back(str);
        }

        if (::GetLastError() == ERROR_INSUFFICIENT_BUFFER)
        {
        }
    }
    return arr;
}

```

### ***Код використання бібліотеки:***

```

void _com_reader() {
    static std::mutex m;

    m.lock();
    std::string s;
    s = COM::SerialRead();
    com_info = s;

    m.unlock();
}
void com_Reader() {
    std::queue<std::future<void>> queue;

    if (queue.empty()) {
        queue.push(std::async(std::launch::async, _com_reader));
    }
    else {
        queue.front().wait();
        queue.pop();
    }
}
}

```