

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Чорноморський національний університет імені Петра Могили**  
**Факультет комп'ютерних наук**  
**Кафедра комп'ютерної інженерії**

ДОПУЩЕНО ДО  
ЗАХИСТУ

Завідувач кафедри,  
д-р техн. наук, проф.

\_\_\_\_\_ І.  
М. Журавська

« \_\_ » \_\_\_\_\_  
2023 р.

**КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА**  
**Програмно-апаратний комплекс виявлення**  
**падіння за технологією IoT на базі ESP8266**

Спеціальність 123 «Комп'ютерна інженерія»

123 – КМР.1 – 605.21710506

**Студент**

\_\_\_\_\_ С. І. Волощук  
*підпис*

« \_\_ » \_\_\_\_\_ 2023 р.

**Керівник** канд. техн. наук, доцент

\_\_\_\_\_ В. Ю. Савінов  
*підпис*

«\_\_» \_\_\_\_\_ 2023 р.

**Миколаїв – 2023**

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	4
ВСТУП	5
1 АНАЛІТИЧНА ЧАСТИНА. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ СИСТЕМИ, ЩО РОЗРОБЛЯЄТЬСЯ. ФОРМУВАННЯ ВИМОГ ДО АПАРАТНО-ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	7
1.1 Контекстно-залежні системи	8
1.2 Носимі пристрої	9
1.3 Виклики	11
1.4 Проблеми	13
1.5 Аналогічні рішення	15
1.6 Вимоги до апаратно-програмного забезпечення	18
Висновки до розділу 1	19
2 МАТЕМАТИЧНІ МЕТОДИ ТА ПРОЄКТУВАННЯ СИСТЕМИ ЩО РОЗРОБЛЯЄТЬСЯ	20
2.1 Принцип роботи акселерометра	20
2.2 Принцип роботи гіроскопа	26
2.3 Метод виявлення падіння	33
Висновки до розділу 2	34
3 АПАРАТНО-ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ	35
3.1 Вибір апаратної платформи	35
3.2 Детальний опис обраної апаратної платформи та її компонентів	36
3.3 Побудова принципової та макетної схеми	41
3.4 З'єднання модуля GY-521 з апаратною платформою	43
3.5 Розробка програмної частини	44
Висновки до розділу 3	53
4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ. АНАЛІЗ РЕЗУЛЬТАТІВ ОБЧИСЛЕНЬ	54
4.1 Виконання налаштувань	54

4.2 Визначення нижньої та верхньої межі прискорення та меж кутової швидкості	59
4.3 Перевірка працездатності	65
4.4 Порівняння споживання до та після оптимізації	66
4.5 Порівняння прототипу з існуючим рішенням	67
Висновки до розділу 4	69
ВИСНОВКИ	70
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	71
ДОДАТОК А Програмний код апаратної частини для виявлення падіння	74

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

АЦ	–	Аналого-цифровий перетворювач
П	–	Контекстно-залежні системи
КЗ	–	Рідкокристалічний
С	–	Activities of Daily Living
PK	–	Controller Area Network
AD	–	Digital Motion Processor
L	–	Electrically Erasable Programmable Read-Only Memory
CA	–	Electronic mail
N	–	General-purpose input/output
DM	–	Global Positioning System
P	–	Inter-Integrated Circuit
EE	–	Integrated development environment
PROM	–	Institute of Electrical and Electronics Engineers
Em		
ail		
GPI		
O		
GP		
S		
I2C		
IDE		
IEE		
E		
IFT	–	If This Then That
TT	–	Internet of Things
IoT	–	Long-Term Evolution
LT		
E		

	ME –	Microelectromechanical systems
MS	–	Secure Digital Input/Output
	SDI–	Sensitivity
O	–	Short Message Service
	SE	
	SM	
S		
	SP –	Specificity
	SPI –	Serial Peripheral Interface
	SR –	Static Random Access Memory
AM	–	Service Set Identifier
	SSI –	Secure Sockets Layer
D		
	SSL	
	TL –	Transport Layer Security
S		
	UA –	Universal Asynchronous Receiver-Transmitter
RT	–	Wireless Fidelity
	Wi-	
Fi		

## ВСТУП

За даними Всесвітньої організації охорони здоров'я [1], приблизно 28-35% людей у віці 65 років і старше щорічно падають, збільшуючись до 32-42% у віці старше 70 років. Частота падінь зростає з віком і ступенем слабкості. Насправді кількість падінь експоненціально збільшується з віковими біологічними змінами, що призводить до високої частоти падінь і пов'язаних з падінням травм у суспільствах, що старіють. Якщо в

найближчому майбутньому не вжити профілактичних заходів, прогнозується, що кількість травм, спричинених падінням, у 2030 році зросте на 100%.

Оскільки населення постійно старіє, існує нагальна потреба в розробці систем виявлення падінь. Завдяки швидкому розвитку сенсорних мереж та Інтернету речей (IoT) [2] взаємодія людини та комп'ютера за допомогою сенсорного синтезу вважається ефективним методом вирішення проблеми виявлення падінь.

**Актуальність магістерської роботи** полягає в тому, що спроектований комплекс може бути корисним для людей похилого віку. Це пов'язано з тим, що він може повідомити людину чи члена сім'ї, коли виявить падіння, і зменшить ризик затримки медичної допомоги.

**Мета магістерської роботи:** спроектувати програмно-апаратний комплекс виявлення падіння з повідомленням користувача.

**Об'єкт дослідження магістерської роботи:** технології та методи виявлення падіння.

**Предмет дослідження магістерської роботи:** програмно-апаратний комплекс для виявлення падіння та відправлення повідомлення про це людині.

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

- проаналізувати матеріал щодо можливих технологій та методи виявлення падіння;
- проаналізувати можливі апаратні платформи для реалізації прототипу;
- змоделювати конструкцію прототипу та підібрати всі необхідні компоненти;
- спроектувати апаратний комплекс для виявлення падіння в реальному часі;

- розробити програмну частину на мові програмування C для отримання даних з датчиків, їх обробки та надсиланні попередження до користувача за допомогою технологій Інтернету речей;
- розробити питання з охорони праці та безпеки життєдіяльності.

**Наукова новизна** полягає в реалізації програмно-апаратного комплексу для виявлення падіння, який відрізняється мінімізацією структури (мінімального використання модулів, що зменшить затрати на побудову комплексу). Також в Україні ще немає патентів на корисну модель для виявлення падіння.

**Практичне значення** отриманих результатів: спроектований комплекс може бути рекомендований для сімей з людьми похилого віку для відстеження можливого падіння та своєчасного повідомлення піклувальника про це.

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

**Публікації.** Основні положення магістерської роботи опубліковані у збірнику матеріалів XXV Всеукраїнської науково-практичної конференції «Могилянські читання–2022» [3].



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

Сьогодні пристрої для виявлення падінь допомагають зменшити ризик довготривалих травм, виявляючи різку зміну положення людини. Пристрій звертається за допомогою, навіть якщо людина не в змозі зробити це самостійно, сигналізуючи про швидку екстрену реакцію для вирішення ситуації.

По суті, структура всіх систем виявлення падіння завжди схожа. Їх головна мета полягає в тому, щоб розрізнити справжнє падіння та повсякденну діяльність (ADL).

Це непросте завдання, оскільки певні ADL, як-от сидючи або переходячи з положення стоячи в положення лежачи, мають велику схожість з падіннями. Таким чином, щоб протестувати детектор падіння, необхідно зібрати дані про падіння та ADL, які можуть бути реальними (дуже важко, особливо для падінь) або змодельованими молодими добровольцями (можливий варіант, прийнятий більшістю авторів). Ці дані реєструються датчиками та можуть бути у формі сигналів прискорення, зображень, сигналів тиску тощо. Потім вони обробляються та класифікуються за допомогою техніки виявлення падінь, яка дозволяє відрізнити падіння від ADL.

У більшості випадків продуктивність детектора виражається через чутливість (SE) і специфічність (SP).

Чутливість – це здатність детектора правильно класифікувати падіння як падіння, тоді як специфічність – здатність детектора правильно класифікувати ADL як ADL [4].

Датчики падіння можна розділити на два типи: контекстно-залежні системи та переносні пристрої.

## 1.1 Контекстно-залежні системи

Ці системи використовують датчики, розміщені в навколишньому середовищі, щоб виявляти падіння. Головна їх перевага полягає в тому, що людині не потрібно носити якийсь спеціальний пристрій. Однак їхня робота обмежена тими місцями, де датчики вже були розміщені раніше [5].

Серед усіх можливих типів датчиків найпоширенішими є камери, датчики підлоги, інфрачервоні датчики, мікрофони та датчики тиску. Системи на основі відео можна розглядати як підкатегорію в цій групі, оскільки вони використовують методи комп'ютерного зору, які відрізняються від решти методів виявлення.

Існує велика варіативність методів виявлення, оскільки вони залежать від типу використовуваного датчика. Усі методи починаються з виділення ознаки, наприклад, співвідношення зросту та ваги людей [6], точок краю від силуету людини, зміни освітленості [7], орієнтації головної осі людини [8], ширини, висоти та глибини людської пози [9], кольору шкіри для виявлення людей [10] тощо. Потім ці характеристики порівнюються та класифікуються, щоб відрізнити звичайну діяльність від реальних падінь за допомогою різних. Наразі досліджено багато різних функцій, і жодна з них не переважає над іншими, оскільки вони дають подібні результати, і порівняння не проводилося.

Питання конфіденційності в контекстно-орієнтованих системах не є другорядними проблемами. Методи захисту конфіденційності залежать від типу використовуваного датчика. У крайньому випадку технології, заснованої на відео, деякі автори вирішили приховати або спотворити зовнішність людини на відео, щоб забезпечити конфіденційність. Хоча конфіденційність слід розглядати вже на стадії проектування, не всі дослідження дотримувалися цього підходу, що є чіткою ознакою того, що деякі системи з урахуванням контексту в основному зосереджені на технологічному розвитку, а не на розгортанні в реальному світі.

В цілому структура всіх методів дуже схожа. Більшість із них починаються з виявлення об'єкта, яке можна виконати через віднімання фону в системах на основі відео, або з інформації, наданої спостереженням датчиків.

Деякі методи також розглядають алгоритм відстеження для фільтрації позиції об'єктів. Потім виділяють деякі ознаки виявлених об'єктів, які повинні мати достатню дискримінаційну силу для ідентифікації подій падіння. Вони використовуються для класифікації подій як падіння або не падіння за допомогою широкого спектру методів: модель суміші Гауса [11], методи на основі правил, фільтрація Байеса, правило найближчого сусіда, приховані Марковські моделі [12], методи визначення порогів, нечітка логіка [13] тощо.

Деякі дослідження підтверджують, що відбулося падіння, виконуючи виявлення бездіяльності після падіння.

## **1.2 Носимі пристрої**

Носимі пристрої можна визначити як мініатюрні електронні пристрої на основі датчиків, які носій носить під одягом, з ним або поверх нього. Переважна більшість носимих детекторів падіння мають форму акселерометрів. Деякі з них також містять інші датчики, такі як гіроскопи, для отримання інформації про положення пацієнта.

Активно досліджується використання застосунків на основі акселерометрів і гіроскопів для оцінки ходи та балансу, оцінки ризику падіння та моніторингу рухливості. Ця тенденція посилилася протягом останніх років через доступність дешевих вбудованих датчиків, що входять до складу смартфонів.

### 1.2.1 Акселерометр, прикріплений до тіла

Дані про прискорення збираються під час падінь за допомогою незалежних трьох-осьових акселерометрів, прикріплених до різних частин тіла.

Застосовувані методи включають машину опорних векторів, оптимізацію роєм частинок перегрупування, розподіл гаусового кластерного знання, багатошаровий перцептрон, наївний Байєс, дерево рішень, ZeroR і OneR. Згідно з Кердегарі та ін. [14], багатошаровий перцептрон є хорошим варіантом під наглядом, хоча не існує стандартизованої методики, яка б була широко прийнята науковою спільнотою.

Ширші категорії типових подій падіння — це вперед, назад і вбік. Що стосується положення акселерометра, місце розташування на талії здається оптимальним для виявлення падіння. Прикріплені на талії акселерометри розташовані поблизу центру ваги тіла, надаючи надійну інформацію про рухи тіла суб'єкта.

Як і в попередній категорії, заявлена продуктивність дуже висока, але справа в тому, що ці пристрої мало використовуються в повсякденній геріатричній практиці та не мають значного промислового впровадження детекторів падіння через значну кількість помилкових тривог, що призводить до невідповідних оповіщення. Заявлені результати дійсні для лабораторних середовищ з обмеженими даними або в обмежених умовах, але в реальних сценаріях є багато неконтрольованих факторів, які призводять до різкої втрати продуктивності. Тому дослідження повинні включати більш тривалі тести та включати показники, ближчі до реального використання, наприклад, кількість помилкових тривог на день, що може дати цільовим користувачам більш реалістичне уявлення про справжню продуктивність системи.

### 1.2.2 Вбудований в смартфон акселерометр

Сучасні смартфони оснащені багатим набором вбудованих датчиків, таких як акселерометр, цифровий компас, гіроскоп, GPS, мікрофон і камера.

Алгоритми низької складності, засновані на порогових значеннях, використовуються в більшості рішень, і лише деякі йдуть далі та застосовують стратегії машинного навчання. Вони використовують опорні векторні машини, розріджену мультиноміальну логістичну регресію, наївну Байєсовську систему, K-найближчі сусіди, дерева рішень і багаторівневі нейронні мережі.

Що стосується положення телефону, то талія залишається кращою частиною тіла, хоча спостерігається тенденція до стегна, що збігається з розташуванням кишені.

Деякі з досліджень призвели до реальних програм для виявлення падінь, які доступні для завантаження в Google Play. Таким чином, у цьому сховищі застосунків було проведено пошук із ключовими словами «детектор падіння» або «виявлення падіння». Загалом було отримано 9 застосунків, з них 7 для літніх людей. Наводячи деякі статистичні дані, 3 з них повідомили про від 1000 до 5000 завантажень, тоді як решта мали менше 500. Хоча ці цифри вказують на певний рівень інтересу, вони все ще далекі від кількості потенційних користувачів. Зосереджуючись на рейтингу програми, було виявлено, що в середньому лише 6 людей висловили свою думку про них. Це ознака того, що люди, які користуються цими програмами, здається, не в захваті від них.

### 1.3 Виклики

Конструкція детекторів падіння стикається з деякими серйозними викликами.

### 1.3.1 Продуктивність у реальних умовах

Датчики падіння повинні бути максимально точними та надійними. Надійна система виявлення падіння повинна демонструвати як високу чутливість, так і специфічність. Це іноді досягається в експериментальних середовищах, але при застосуванні до реальної ситуації швидкість виявлення знижується.

Ці пристрої розроблені та перевірені в контрольованих умовах, наприклад, вони використовують дані про падіння та ADL молодих людей, змодельовані на розсуд кожного автора через відсутність стандартизованої процедури або загальнодоступної бази даних для порівняння.

Крім того, варто зазначити, що датчики падіння орієнтовані на людей похилого віку, тому вони також повинні бути залучені до їх розробки. Користувачі повинні носити пристрої довше (принаймні місяці).

### 1.3.2 Зручність використання

Детектори падіння на основі смартфонів є привабливими через широке використання телефонів, навіть серед літніх людей. Однак згідно з більшістю досліджень, смартфони ставлять у стандартизовану позицію. Це дозволило провести дуже стереотипні вимірювання, які допомогли оцінити точність, але зробили результати менш застосовними до того, як люди щодня носять свої смартфони (наприклад, у кишенях чи сумочках).

Майбутні детектори на основі смартфонів не повинні обмежувати розміщення пристрою однією частиною тіла (поперек, зап'ястя, груди тощо). Смартфони слід використовувати у звичайний спосіб, без обмежень щодо їх розташування чи функцій. Це може призвести до зниження рівня виявлення.

### 1.3.3 Простота

Складність пристрою створює серйозну проблему для людей похилого віку, оскільки вони можуть бути не знайомі з електронними пристроями.

Щоб подолати цю проблему, важливе значення має спосіб функціонування системи.

Детектор має активуватися та працювати автоматично, без втручання користувача. Системи бачення, як і інші неінтрузивні методи, дуже хороші в цьому сенсі. Однак деякі переносні пристрої, такі як смартфони, мають інші переваги, які можуть допомогти покращити сприйняття детекторів падіння. Вони можуть працювати як у приміщенні, так і на відкритому повітрі та інтегрувати в один пристрій не лише функцію виявлення падіння, але й інші програми охорони здоров'я.

Таким чином можна було б подолати традиційне небажання носити різні пристрої, кожен з яких має певну функцію. Проте використання смартфонів літніми людьми не без труднощів: ці пристрої, як вони були задумані, є для них основною перешкодою для зручності використання. Доказом цього є відсутність рейтингу в існуючих програмах виявлення падіння, що є симптомом малого реального використання.

У цьому сенсі потрібні потенційні рішення для покращення зручності використання та доступності смартфонів. Датчики падіння високо цінуються людьми похилого віку, які продемонстрували позитивне ставлення до рішень на основі смартфонів після практичної демонстрації кількох допоміжних технологій.

Цей висновок узгоджується з роботою Плаза та ін. [15], який розглянув мобільні додатки для людей похилого віку та виявив, що вони найчастіше пов'язані з двома сферами: «Health – wellness – home care» і «Safety – security – mobility».

#### **1.4 Проблеми**

Існують деякі проблеми, які можуть перешкоджати продуктивності системи. З них можна виокремити: обмеження смартфона, конфіденційність, порівняння різних методів з публічних джерел, реальні падіння.

#### 1.4.1 Обмеження смартфона

Тенденція до детекторів на основі смартфонів створює деякі проблеми. По-перше, смартфони не є пристроями, спочатку призначеними для виявлення падіння або будь-якої іншої важливої для безпеки програми.

Можуть виникнути труднощі з операціями в реальному часі, архітектурою датчиків, стабільністю частоти дискретизації акселерометра, особливостями операційної системи тощо. Дійсно, той самий детектор падіння може поводитися дещо по-різному залежно від моделі смартфона, у якому він працює. встановлено. Цю можливість слід враховувати в реальному сценарії.

По-друге, смартфони не можна перевантажувати безперервними датчиками, які погіршують продуктивність телефону, наприклад, виснажуючи заряд батареї. Важливо керувати циклом сну сенсорних компонентів, щоб компенсувати кількість споживаного заряду батареї. Тим не менш, час автономної роботи смартфона завжди низький, що може перешкодити його прийняттю. Це не дрібна проблема, особливо враховуючи, що система призначена для літніх людей з обмеженими можливостями пересування.

По-третє, потрібні прості у використанні смартфони, і тут ми в руках виробників. Потенційний ринок додатків для людей з низькими технічними навичками вплине на розвиток адаптованих пристроїв. Тим не менш, детектори падіння навряд чи досягнуть у найближчому майбутньому надійності та стабільності, досягнутої іншими допоміжними технологіями, такими як пристрої для натискання на допомогу.

#### 1.4.2 Конфіденційність

Проблеми конфіденційності систем на основі датчиків і детекторів падіння були гарячою темою. Звичайно, не всі типи датчиків однаково вразливі: КЗС загалом і системи на основі зору зокрема набагато більше



схильні до проблем конфіденційності, ніж, наприклад, пристрої, що носяться на тілі, засновані на прискоренні.

У будь-якому випадку має бути гарантований захист конфіденційних контекстних даних. Проблеми конфіденційності не повинні перешкоджати потенційним перевагам допоміжних технологій, оскільки в той же час конфіденційністю не можна пожертвувати заради отримання інших переваг. Загалом у дослідженнях виявлення падіння зазвичай бракує стратегій забезпечення конфіденційності даних. Це свідчить про те, що вони ще далекі від реального розгортання.

#### 1.4.3 Порівняння різних методів з публічних джерел

Порівнювати різні підходи складно, тому що кожен автор отримує дані різними способами: типи змодельованих падінь і ADL, положення детектора, частота дискретизації, часова довжина сигналу, витягнуті характеристики тощо.

Дослідження слід зосереджувати не лише на алгоритмі, а для використання, а також на шляху отримання та обробки сигналів перед подачею на класифікатор. Загальнодоступна база даних із сигналами акселерометра та відео падіння людей може допомогти порівняти різні методи та покращити відтворюваність результатів. Спільне використання вихідного коду алгоритмів також було б цінним варіантом.

#### 1.4.4 Реальні падіння

Більшість досліджень використовують дані симуляції ADL і падінь молодих або зрілих людей. Навіть якщо припустити, що вони були оприлюднені для справедливого порівняння різних методів, незрозуміло, чи змодельована діяльність є репрезентативною для своїх реальних аналогів.

Оскільки неприйнятно піддавати літніх людей симуляції падіння, оцінка детекторів сильно обмежена. Лише деякі дослідження надають дані про

прискорення реальних падінь людей похилого віку, але кількість зареєстрованих подій залишається низькою.

Крім того, механізми падінь невідомі, оскільки вони не можуть бути точно задокументовані .

## 1.5 Аналогічні рішення

На ринку існує чимала кількість приладів в яких вбудована функція виявлення падіння, але більшість з них має високу ціну, або працює тільки в окремих країнах.

### 1.5.1 Home and Wellness – Reliable Fall Detection Medical Alert Device

Дане рішення має вбудований сенсор виявлення падіння, GPS та LTE модулі для швидкого надання координатів медичному закладу та вбудований мікрофон для виклику допомоги.

Даний прилад працює виключно на території Сполучених Штатів Америки, та просить щомісячно 35\$ передплати за користування приладом.

На рис. 1.1 можна побачити прилад для виявлення падіння від компанії Home and Wellness.



## Рисунок 1.1 – Home and Wellness Fall Detection Medical Alert Device

### 1.5.2 Galaxy On the Go Mobile System with GPS & Fall Detection

На рис. 1.2 можна побачити прилад для виявлення падіння від компанії Galaxy Medical Alert Systems.



### Рисунок 1.2 – Galaxy On the Go Mobile System with GPS & Fall Detection

Дане рішення базується на трьохосьовому акселерометру з власними розробленими алгоритмами виявлення падіння. В наявності GPS, WiFi-модулі дані з яких оновлюються кожні 30 секунд після натискання кнопки Help, а також LTE-модуль для зв'язку з медичним закладом. Ресурс девайсу – до 96 годин постійної роботи. Працює виключно на території Канади з щомісячною передплатою від 50\$.

Розробник зауважує, що даний пристрій не виявить 100% усіх падінь. Завжди потрібно натискати кнопку на девайсі, щоб отримати допомогу якомога швидше. Прилад був розроблений та рекомендований виробником для носіння на зовнішній частині одягу для правильного використання системи виявлення падіння.

### 1.5.3 Apple Watch SE 2

Apple Watch SE 2 оснащено функцією виявлення падіння. Цю функцію потрібно налаштувати за допомогою програми Watch на iPhone.

Після ввімкнення Apple Watch SE 2 може виявити сильне падіння та автоматично повідомити служби екстреної допомоги для подальшої допомоги. Якщо падіння сталося, годинник він подасть звуковий сигнал і відобразить сповіщення. Якщо людина продовжує рухатись, годинник не буде автоматично викликати екстрену службу і чекатиме вашої відповіді. Однак, якщо людина не рухається більше хвилини, годинник починає 30-секундний відлік, вібруючи та лунаючи звуковий сигнал, який поступово стає голоснішим. Після закінчення зворотного відліку він автоматично сповістить екстрені служби та повідомить поточне місцезнаходження.

На рис. 1.3 продемонстровано зовнішній вигляд Apple Watch SE 2.



Рисунок 1.3 – Смарт-годинник Apple Watch SE 2

Вартість даного пристрою від 300\$. З мінусів можна виокремити, що для роботи смарт годинника його потрібно активувати тільки за допомогою iPhone з операційною системою від iOS 12.

### **1.6 Вимоги до апаратно-програмного забезпечення**

Після аналізу доступної інформації, було сформовані певні вимоги до майбутнього апаратно-програмного забезпечення:

- Невеликий розмір для зручного носіння;
- простота використання;
- велика точність виявлення падіння;

- розмір програмної частини не повинен перевищувати ліміт платформи;
- низька ціна у порівнянні з аналогами.

### **Висновки до розділу 1**

Під час оформлення розділу 1, було опановано та проаналізовано матеріал щодо різних систем, технологій та методів виявлення падіння.

Було виокремлено два типи датчиків падіння: контекстно-залежні системи та носимі пристрої. До контекстно-залежних систем відносять камери, датчики підлоги, інфрачервоні датчики, мікрофони та датчики тиску. До носимих пристроїв відносять акселерометр прикріплений до тіла, або акселерометр вбудований в смартфон чи смарт-годинник.

Щодо проблем, можна виділити проблеми через обмеження смартфона та проблеми конфіденційності систем. Через обмеження смартфона можуть виникнути труднощі з операціями в реальному часі, архітектурою датчиків, стабільністю частоти дискретизації акселерометра, особливостями операційної системи тощо. Проблеми конфіденційності не повинні перешкоджати потенційним перевагам допоміжних технологій, оскільки в той же час конфіденційністю не можна пожертвувати заради отримання інших переваг.

Були проаналізовані аналоги, та сформовані вимоги до майбутнього апаратно-програмного забезпечення. Апаратно-програмний комплекс повинен бути: невеликого розміру для зручного носіння, простим у використанні, точним при розрахунках для виявлення падіння, розмір програмної частини не повинен перевищувати ліміт платформи, дешевшим у порівнянні з аналогами.

## 2 МАТЕМАТИЧНІ МЕТОДИ ТА ПРОЄКТУВАННЯ СИСТЕМИ ЩО РОЗРОБЛЯЄТЬСЯ

### 2.1 Принцип роботи акселерометра

Акселерометри – це електромеханічні пристрої які вимірюють прискорення, тобто рівень зміни швидкості. На рис. 2.1 можна побачити 9-осьовий акселерометр MPU-9250.

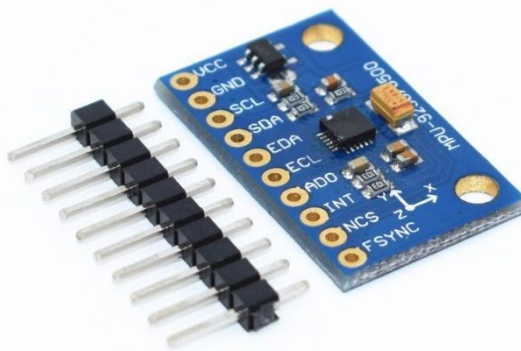


Рисунок 2.1 – Акселерометр MPU-9250

Прискорення, з математичної точки зору, є векторною фізичною величиною, першою похідною швидкості по часу та за величиною дорівнює зміні швидкості за одиницю часу. Це поняття можна зрозуміти на прикладі механізму звичайного акселерометра, який складається з маси, прикріпленої до пружини. Маса може вільно рухатися в одному або декількох напрямках, а пружина діє з силою, пропорційною зміщенню маси від початкового положення. Переміщення маси вимірюється за допомогою ємнісного, п'єзоелектричного або п'єзорезистивного датчика, а отриманий електричний сигнал використовується для обчислення прискорення.

Формула для визначення прискорення за допомогою акселерометра має вигляд:

$$a = \frac{(f-kx)}{m}, \quad (1)$$

де  $a$  – прискорення;  
 $f$  – прикладена сила;  
 $k$  – константа пружини;  
 $x$  – переміщення маси;  
 $m$  – маса акселерометра.

Важливо визнати, що на вимірювання акселерометра можуть впливати зміщення та масштабні коефіцієнти, які необхідно визначати в процесі калібрування. Зсуви відносяться до постійних значень вимірювань акселерометра, тоді як масштабні коефіцієнти є мультиплікативними константами, які перетворюють вимірювання акселерометра в потрібні одиниці виміру. Вони можуть бути встановлені за допомогою різних методів калібрування.

Одним з таких методів є метод  $g$ -чутливості, який використовується для визначення масштабних коефіцієнтів акселерометра. У цьому підході акселерометр поміщають у відоме гравітаційне поле і вимірюють його вихідний сигнал. Масштабні коефіцієнти можна обчислити, порівнюючи вимірний вихід з відомим гравітаційним полем, використовуючи наступну формулу:

$$scale\_factor = \frac{g_{\text{вимірне}}}{g_{\text{очікуване}}}, \quad (2)$$

$g_{\text{вимірне}}$  – вимірне прискорення (у  $m/s^2$ ) у відомому гравітаційному полі;

$g_{\text{очікуване}}$  – очікуване прискорення (у  $m/s^2$ ) у цьому відомому гравітаційному полі.

Іншим популярним методом калібрування є метод зміщення в невагомості, який використовується для визначення зміщення акселерометра. У цьому методі акселерометр поміщають у середовище невагомості і вимірюють його вихідні дані. Потім зміщення обчислюється як середнє значення за певний проміжок часу за наступною формулою:

$$bias_x = \left(\frac{1}{N}\right) * \sum x_i \quad (3)$$

### 2.1.1 MEMS акселерометр

Акселерометр MEMS [22] – це мікромеханічна структура, побудована на кремнієвій пластині.

Одним із найбільш часто використовуваних MEMS-акселерометрів є акселерометр ємнісного тип. Ємнісний акселерометр MEMS відомий своєю високою чутливістю та точністю при високих температурах. Пристрій не змінює значення залежно від використовуваних основних матеріалів і залежить лише від ємнісного значення, яке виникає через зміну відстані між пластинами.

Якщо дві пластини розташовані паралельно одна одній і розділені відстанню «d», і якщо «ε» є діелектричною проникністю матеріалу, що розділяє, то отриману ємність можна записати як:

$$C_0 = \frac{\varepsilon A}{d}, \quad (4)$$

$$\varepsilon A = \varepsilon_0 \varepsilon A, \quad (5)$$

A – площа електродів.

Зміна значень ε, A або d допоможе знайти зміну ємності та, таким чином, допоможе в роботі перетворювача MEMS. Значення акселерометра в основному залежать від зміни значень d або A.

Типовий MEMS-акселерометр показаний на рис 2.2. Його також можна назвати простим одноосьовим акселерометром. Якщо більше наборів конденсаторів зберігаються під кутом 90 градусів один до одного, можна створити 2- або 3-осьовий акселерометр. Простий перетворювач MEMS в основному складається з рухомої мікроструктури або пробної маси, яка з'єднана з механічною системою підвіски, а отже, з системою відліку.



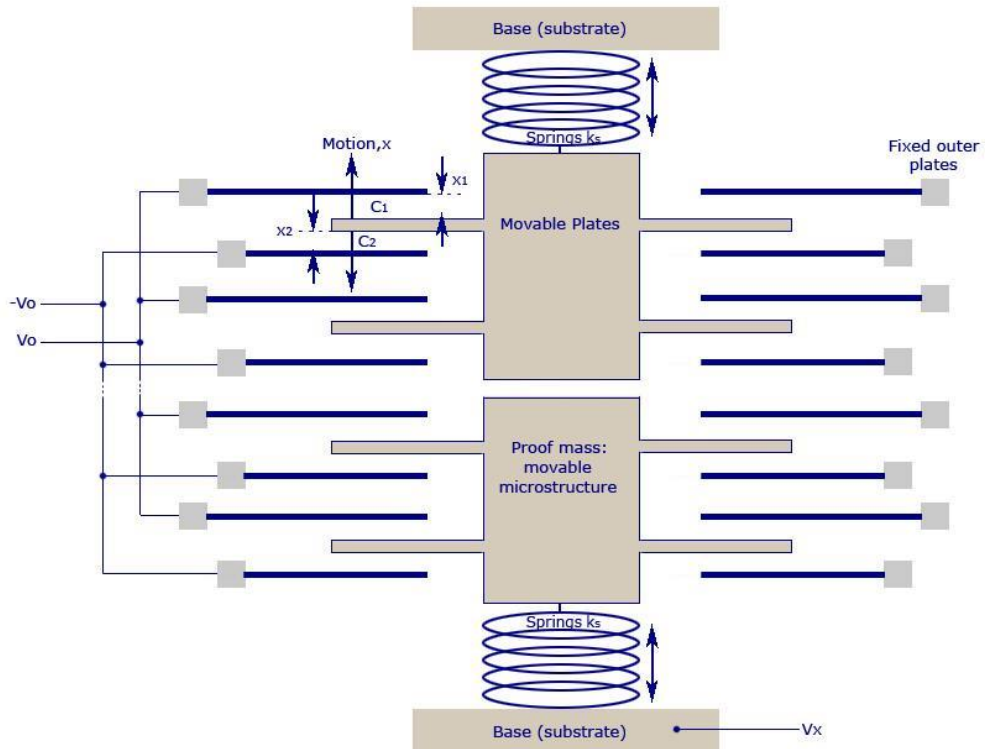


Рисунок 2.2 – Простий одноосьовий MEMS акселерометр

Рухомі пластини та нерухомі зовнішні пластини діють як пластини конденсатора. Коли застосовується прискорення, доказова маса рухається відповідно. Це створює ємність між рухомою та нерухомою зовнішніми пластинами.

Коли застосовується прискорення, відстань між двома пластинами зміщується як  $X_1$  і  $X_2$ , і вони виявляються функцією виробленої ємності.

Із зображення вище видно, що всі датчики мають кілька наборів конденсаторів. Усі верхні конденсатори з'єднані паралельно, щоб отримати загальну ємність  $C_1$ , а нижні – загальну ємність  $C_2$ .

Якщо  $V_x$  — це вихідна напруга пробної маси, а  $V_0$  — вихідна напруга, створювана між пластинами, тоді:

$$(V_x + V_0)C_1 + (V_x - V_0)C_2 = 0 \quad (6)$$

На рис 2.3 показано схему, яка використовується для обчислення прискорення через зміну відстані між пластинами конденсатора. Результат, отриманий для різних значень прискорення, також показано графічно.

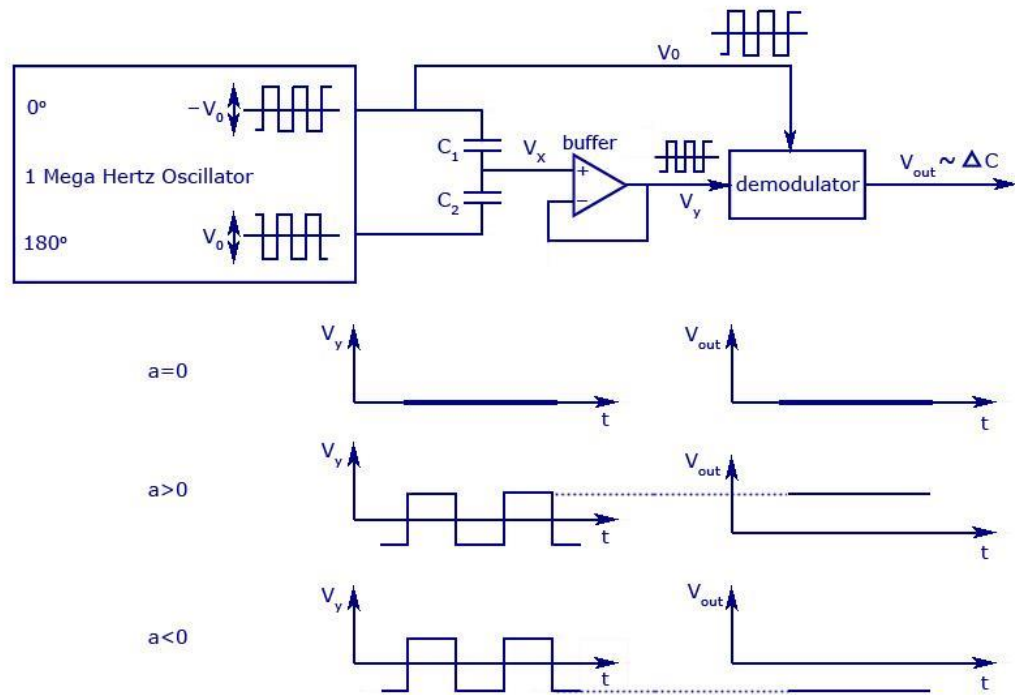


Рисунок 2.3 – Процес обчислення прискорення

Якщо прискорення не задано ( $a=0$ ), вихідна напруга також буде нульовою.

Коли задано прискорення, наприклад ( $a>0$ ), значення  $V_x$  змінюється пропорційно значенню  $V_0$ .

Коли задано уповільнення, наприклад ( $a<0$ ), сигнали  $V_x$  і  $V_y$  стають негативними. Демодулятор створює вихідний сигнал, що дорівнює знаку прискорення, оскільки він множить обидва значення  $V_y$  і  $V_0$  для отримання  $V_{OUT}$ , який має правильний знак прискорення та правильну амплітуду.

Довжина відстані,  $d$  і вага тестової маси дуже малі. Пробна маса важить не більше 0,1 мікрограма, вихідна ємність становить приблизно 20 аФ, а відстань між пластинами — не більше 1,3 мікрометра.

Вибирати пристрій потрібно з урахуванням його шумових характеристик. Якщо необхідно визначити значення прискорення за умов низької сили тяжіння, характеристики шуму можуть легко вплинути на його точність. Кажуть, що MEMS-акселерометр має три параметри, що створюють

шум: від схеми формування сигналу, від вібрацій, створюваних пружинами, і від системи вимірювання вихідного сигналу.

### 2.1.2 Переваги та недоліки MEMS акселерометрів

Переваги використання акселерометра MEMS включають:

- низьку вартість: акселерометри MEMS є відносно недорогими у виробництві, що робить їх економічно ефективним рішенням для багатьох застосувань;
- невеликий розмір: акселерометри MEMS дуже малі за розміром, що робить їх ідеальними для використання в портативних і компактних пристроях;
- високу чутливість: акселерометри MEMS мають високу чутливість і можуть виявляти дуже незначні зміни прискорення;
- широкий діапазон застосувань: MEMS-акселерометри можна використовувати в широкому діапазоні застосувань, таких як смартфони, транспортні засоби, промислове обладнання та медичні пристрої;
- високу надійність: акселерометри MEMS дуже надійні та мають тривалий термін служби.

До недоліків використання акселерометра MEMS належать:

- обмежений діапазон вимірювання: діапазон вимірювання акселерометра MEMS зазвичай обмежується кількома g (прискорення сили тяжіння), чого може бути недостатньо для деяких застосувань;
- шум і дрейф: на акселерометри MEMS можуть впливати шум і дрейф з часом, що може знизити їх точність і стабільність;
- обмежений діапазон температур: діапазон робочих температур акселерометра MEMS зазвичай обмежений, що може вплинути на його роботу в екстремальних умовах;

- чутливість до вібрації: на акселерометри MEMS можуть впливати вібрація та інші фактори навколишнього середовища, що може знизити їх точність і стабільність.

Загалом акселерометри MEMS є привабливим варіантом для багатьох додатків через їх низьку вартість, малий розмір і високу чутливість, але на їх продуктивність і надійність можуть впливати шум, дрейф і фактори навколишнього середовища. Важливо враховувати ці фактори при виборі MEMS-акселерометра для конкретного застосування.

## 2.2 Принцип роботи гіроскопа

Вимірювання кутової швидкості, також відомої як швидкість зміни кутового положення, є важливим аспектом у галузі фізики. Гіроскоп є найбільш поширеним пристроєм для цієї мети, який працює за принципом колеса, що обертається, або ротора, підвішеного в рамці. Кутова швидкість пристрою визначається шляхом виявлення прецесії, або коливання, дзиги. Зазвичай це досягається завдяки використанню ємнісного, п'єзоелектричного або п'єзорезистивного чутливого механізму, а отриманий електричний сигнал обробляється для обчислення кутової швидкості. На рис 2.4 можна побачити побудову гіроскопа.



Рисунок 2.4 – Будова гіроскопа

Математичне представлення обчислення кутової швидкості за допомогою гіроскопа представлено рівнянням:

$$\omega = I^{-1} * (T - D), \quad (7)$$

де  $\omega$  – кутова швидкість (рад/с);

$I$  – момент інерції ротора;

$T$  – крутний момент, прикладений до ротора;

$D$  – демпфуючий момент.

Як і акселерометри, гіроскопи також стикаються зі зміщеннями та масштабними коефіцієнтами, які необхідно компенсувати за допомогою калібрування. Зсуви – це зміщення у вимірюваннях гіроскопа, тоді як масштабні коефіцієнти – це мультиплікативні константи, які переводять вимірювання в потрібні одиниці. Для усунення похибок, використовуються методи калібрування.

Одним із методів калібрування є метод статичного зсуву, який передбачає утримання гіроскопа нерухомим і вимірювання його вихідного сигналу протягом певного періоду, обчислює зсув як середній вихідний сигнал за цей час за формулою:

$$bias_x = \left(\frac{1}{N}\right) * \sum x_i \quad (8)$$

Іншим методом калібрування є метод динамічного зсуву, коли гіроскоп піддається впливу відомої кутової швидкості, обчислюється зсув, порівнюючи вимірний вихід з очікуваною кутовою швидкістю за формулою:

$$bias_x = (x_{measured} - x_{expected}) \quad (9)$$

Важливо розуміти, що точність калібрування акселерометра і гіроскопа залежить від припущення, що пристрій функціонує оптимально, а середовище, в якому він перебуває, відоме і контрольоване. Щоб забезпечити безперервну роботу, рекомендується повторювати процес калібрування через регулярні проміжки часу.

### 2.2.1 MEMS гіроскоп

Гіроскоп MEMS [23] – це прилад, призначений для вимірювання кутової швидкості об'єкта, використовуючи принцип обертального моменту. Зазвичай він складається з мініатюрної підвішеної маси, яка може обертатися

навколо однієї або декількох осей. Обертання маси виявляється і кількісно вимірюється за допомогою різних датчиків, таких як ємнісні або п'єзоелектричні датчики, які потім обробляються електронними схемами для отримання точного вимірювання кутової швидкості. Основний принцип роботи гіроскопа MEMS полягає в наступному:

У центр пристрою поміщають невелику підвішену масу. Маса зазвичай виготовляється з кремнієвої пластини і має форму диска або кільця.

Маса підвішена кількома тонкими, гнучкими балками, які називаються «доказовими масами». Ці балки прикріплені до основи і дозволяють масі обертатися навколо однієї або кількох осей. Балки підвіски сконструйовані таким чином, щоб мати певну постійну пружини, яка визначає власну частоту коливань пробних мас.

Коли до пристрою прикладається кутова швидкість, пробні маси відхиляються внаслідок ефекту Коріоліса, який спричинений кутовою швидкістю, що діє на масу, і викликає дію сили, перпендикулярної до напрямку руху. Ця сила змушує пробні маси відхилитися, створюючи вимірне зміщення. Величина переміщення прямо пропорційна кутовій швидкості пристрою.

Для вимірювання прогину пробних мас використовуються різні типи датчиків. Найбільш поширеними датчиками, які використовуються в MEMS-гіроскопах, є ємнісні та п'єзоелектричні датчики. Ємнісні датчики вимірюють переміщення пробних мас, виявляючи зміни ємності між пробними масами та електродами на підкладці. З іншого боку, п'єзоелектричні датчики перетворюють механічне зміщення пробних мас в електричний сигнал.

Потім вихідні дані датчиків обробляються електронними схемами для забезпечення вимірювання кутової швидкості. Електронні схеми, які обробляють сигнал, підсилюють, фільтрують і уточнюють дані, відіграють вирішальну роль у поліпшенні співвідношення сигнал/шум і усуненні будь-

яких небажаних шумів і перешкод. Крім того, вихід датчика повинен бути відкалібрований і компенсований для різних неідеальних параметрів, таких як температура, зміщення, перехресна чутливість і масштабний коефіцієнт.

Фундаментальним принципом, що лежить в основі функціонування MEMS-гіроскопа, є ефект Кориоліса [24]. Це явище, в якому об'єкт, що рухається в системі відліку, що обертається, відчуває перпендикулярну силу через обертання системи відліку, використовується в MEMS-гіроскопах для створення вимірюваного переміщення тестових мас.

Сила Кориоліса може бути виражена математично у вигляді відомого рівняння:

$$F_c = 2 * m * (\omega \times v), \quad (10)$$

де  $F_c$  – сила Кориоліса;

$m$  – маса пробних мас;

$\omega$  – кутова швидкість пристрою;

$v$  – швидкість пробних мас.

На рис. 2.5 можна побачити графічно принцип роботи вібраційного гіроскопа MEMS.

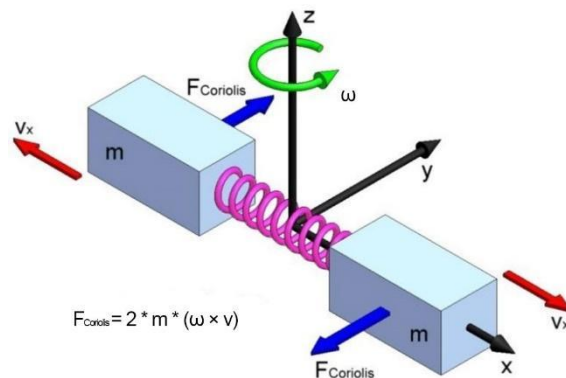


Рисунок 2.5 – Принцип роботи вібраційного гіроскопа MEMS

Математичне представлення роботи MEMS-гіроскопа зазвичай виводиться з рівнянь руху тестових мас. Ці рівняння моделюються як прості гармонічні осцилятори, а кутова швидкість пристрою обчислюється на основі відхилення тестових мас. Розрахунок виконується за допомогою математичної формули.:

$$\omega = \left(\frac{k}{I}\right) * \delta, \quad (11)$$

де  $\omega$  – кутова швидкість

$k$  – пружність пробних мас

$I$  – момент інерції пробних мас

$\delta$  – прогин пробних мас

Режими роботи гіроскопа з мікроелектромеханічними системами визначаються типом кутового руху, для кількісної оцінки якого він призначений. Три найпоширеніші режими – це режими крену, повороту та нахилу.

У режимі крену пристрій спеціально призначений для вимірювання кутової швидкості навколо осі крену, яка є віссю, що проходить паралельно пристрою і перпендикулярна до землі (рис. 2.6). Цей режим в основному використовується в тих випадках, коли пристрій повинен вимірювати обертання навколо своєї поздовжньої осі, наприклад, в літаках і ракетах.

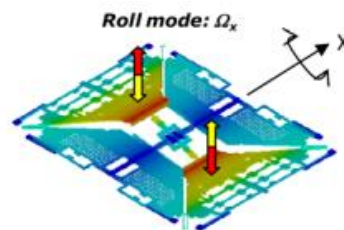


Рисунок 2.6 – Режим крену

У режимі повороту гіроскоп призначений для вимірювання кутової швидкості навколо осі ристання, яка проходить перпендикулярно до пристрою і паралельно до землі (рис. 2.7). Цей режим використовується переважно в тих випадках, коли потрібно виміряти обертання навколо вертикальної осі, наприклад, у транспортних засобах і кораблях.

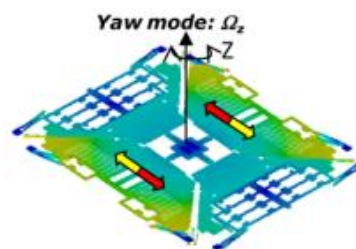


Рисунок 2.7 – Режим повороту



У режимі нахилу пристрій призначений для вимірювання кутової швидкості навколо осі ухилу, яка є паралельною до пристрою і паралельною до землі (рис. 2.8). Цей режим переважно використовується в тих випадках, коли пристрій повинен вимірювати обертання навколо своєї бічної осі, наприклад, в літаках і ракетах.

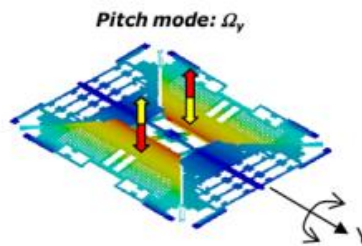


Рисунок 2.8 – Режим нахилу

Крім того, існують MEMS-гіроскопи, які можуть працювати в декількох режимах одночасно, так звані багатоосьові гіроскопи. Ці пристрої застосовуються в ситуаціях, коли потрібно кількісно виміряти кутову швидкість по декількох осях.

Важливо відзначити, що існує багато варіацій і різних типів гіроскопів MEMS з різними конструкціями та принципами роботи. У деяких типах гіроскопів MEMS використовуються кілька пробних мас, розташованих у різних конфігураціях, наприклад, «двосторонній камертон» або «тривісна» конструкція. Ці конструкції можуть забезпечити покращену продуктивність і кращу стійкість до зовнішніх перешкод. Крім того, деякі гіроскопи MEMS використовують різні типи датчиків, наприклад оптичні або магнітні датчики, для виявлення відхилення пробних мас. А також математичні моделі та формули, які використовуються для опису роботи MEMS-гіроскопа, можуть бути досить складними та включати передові концепції фізики та техніки.

### 2.2.2 Переваги та недоліки MEMS гіроскопів

Переваги використання гіроскопа MEMS включають:

- невеликий розмір і низька вартість: гіроскопи MEMS набагато менші та дешевші, ніж традиційні механічні гіроскопи, що робить їх ідеальними для використання в широкому діапазоні застосувань;
- висока точність: MEMS-гіроскопи здатні забезпечувати високоточні вимірювання кутової швидкості, що робить їх придатними для використання в таких програмах, як навігація, системи керування та стабілізація зображення;
- низьке енергоспоживання: MEMS-гіроскопи споживають дуже мало енергії, що робить їх придатними для використання в пристроях, що живляться від батарейок;
- висока надійність: гіроскопи MEMS дуже надійні та мають тривалий термін служби, що робить їх ідеальними для використання в програмах, де простої неприйнятні;
- універсальність: MEMS-гіроскопи можна використовувати в широкому діапазоні застосувань, включаючи навігацію, системи керування, стабілізацію зображення та багато інших.

До недоліків гіроскопів MEMS можна віднести:

- чутливість до змін температури та тиску: MEMS-гіроскопи чутливі до змін температури та тиску, що може вплинути на їх роботу;
- обмежений діапазон вимірювань: MEMS-гіроскопи мають обмежений діапазон вимірювань, що означає, що вони не підходять для використання в програмах, які потребують широкого діапазону вимірювань кутової швидкості;
- чутливість до вібрації: гіроскопи MEMS чутливі до вібрації, що може вплинути на їх продуктивність і точність;

- обмежена точність: MEMS-гіроскопи мають обмежену точність порівняно з іншими типами гіроскопів, особливо при вимірюванні низьких частот або низької амплітуди;
- сприйнятливість до електромагнітних перешкод: гіроскопи MEMS чутливі до електромагнітних перешкод, які можуть вплинути на їх роботу.

### 2.3 Метод виявлення падіння

Виявлення падіння – це використання технологій для виявлення випадків, коли людина впала і може потребувати допомоги. Реалізація такого виявлення часто передбачає інтеграцію даних з акселерометра і гіроскопа. Акселерометр використовується для кількісної оцінки лінійного прискорення, тоді як гіроскоп вимірює кутову швидкість.

Інформація, отримана з акселерометра, використовується для обчислення величини вектора прискорення, який потім використовується для ідентифікації подій падіння. Цей розрахунок здійснюється за допомогою рівняння:

$$a_{magnitude} = \sqrt{a_x^2 + a_y^2 + a_z^2}, \quad (12)$$

де  $a_x, a_y, a_z$  – лінійне прискорення, виміряне по осях x, y та z відповідно.

Встановивши порогове значення для величини вектора прискорення, можна виявити подію падіння, коли ця величина перевищує порогове значення.

Аналогічно, дані з гіроскопа використовуються для визначення кутової швидкості, яка потім дозволяє визначити позу. Обчислення кутової швидкості здійснюється за допомогою рівняння:

$$\omega = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}, \quad (13)$$

де  $\omega_x, \omega_y, \omega_z$  – кутова швидкість, виміряна по осях x, y та z відповідно.

Встановивши порогове значення для кутової швидкості, можна визначити положення людини як вертикальне або лежаче, залежно від того,

чи перевищує кутова швидкість порогове значення, чи опускається нижче нього.

Об'єднання даних з акселерометра і гіроскопа призводить до більш точної системи виявлення падінь зі зменшеною кількістю помилкових тривог. Таке об'єднання можна здійснити за допомогою різних методів, таких як фільтр Калмана або комплементарний фільтр.

Важливо визнати, що на виявлення падіння може впливати кілька факторів, таких як поверхня, на яку падає людина, положення її тіла та фізичний стан. Тому важливо піддавати методи виявлення падінь тестуванню і перевірці в різних сценаріях, щоб гарантувати їхню точність і надійність.

## **Висновки до розділу 2**

Під час оформлення розділу 2, було опановано та проаналізовано матеріал щодо методів та принципів роботи акселерометрів та гіроскопів а також їх різновидностей MEMS акселерометрів та гіроскопів.

Акселерометр MEMS – це мікромеханічна структура, побудована на кремнієвій пластині для вимірювання прискорення. Одним із найбільш часто використовуваних MEMS-акселерометрів є акселерометр ємнісного тип.

До переваг використання акселерометра MEMS можна віднести: низьку вартість, невеликий розмір, високу чутливість, широкий діапазон застосувань, високу надійність. А до недоліків: обмежений діапазон вимірювання, шум і дрейф, обмежений діапазон температур, чутливість до вібрації.

Гіроскоп MEMS – це пристрій, який використовує принцип кутового моменту для вимірювання кутової швидкості об'єкта.

До переваг використання гіроскопа MEMS можна віднести: невеликий розмір і низьку вартість, високу точність, низьке енергоспоживання, високу надійність, універсальність. Недоліки включають: чутливість до змін температури та тиску, обмежений діапазон вимірювань, чутливість до вібрації, обмежену точність, сприйнятливості до електромагнітних перешкод.

Наприкінці було проаналізовано метод та математичні рівняння для виявлення падіння за допомогою MEMS гіроскопів та акселерометрів. Поєднання даних з акселерометра і гіроскопа можна використовувати для більш точного виявлення подій падіння з меншою кількістю помилкових тривог.

## 3 АПАРАТНО-ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

### 3.1 Вибір апаратної платформи

Судячи з вимог до апаратної платформи описаних у розділі 1, а саме: невеликий розмір для зручного носіння, зручність та зрозумілість у використанні, низька ціна у порівнянні з аналогами, було виокремлено декілька можливих апаратних платформ.

#### 3.1.1 Порівняння апаратних платформ

У табл. 3.1 наведено порівняння можливих апаратних платформ.

Таблиця 3.1 – Порівняння апаратних платформ

Характеристика	LoLin NodeMCU v3 ESP8266	ESP-WROOM-32 DevKit V1	Raspberry Pi Pico W	Arduino Nano
MCU	Tensilica L106 32-біт	Tensilica Xtensa LX6	RP2040 з двоядерним процесором ARM Cortex M0+	8-бітний ATmega328P
Частота	80 – 160 МГц	240 МГц	133 МГц	16 МГц
Живлення	3,3 В	3,3 В	5 В	5 В
Вхідна напруга	7 – 12 В	5 – 14 В	1,8 – 5,5 В	7 – 12 В
Пам'ять SRAM	64 КБ	520 КБ	264 КБ	2 КБ
Пам'ять FLASH	До 4 МБ	До 4 МБ	До 2 МБ	32 КБ
Пам'ять EEPROM	4 КБ	4 КБ	4 КБ	1 КБ
Бездротовий зв'язок	802.11 b/g/n	802.11 b/g/n, Bluetooth 4.2	802.11 b/g/n, Bluetooth 5.2	Немає
GPIO	17	36	26	20
Характеристика	NodeMCU v3 ESP8266	ESP-WROOM-32 DevKit V1	Raspberry Pi Pico W	Arduino Nano
UART/SPI/I2C/CAN/SDIO	2/1/1/0/0	3/4/2/1/0	2/2/2/0/0	1/1/1/0/0
Діапазон АЦП	0 – 3,3 В	0 – 3,3 В	0 – 3,3 В	0 – 5 В

Струм споживання	130 мА	80 мА	93 мА	50 мА
Робочий діапазон температур	Від -40°C до 85°C	Від -40°C до 125°C	Від -20°C до 70°C	Від 0°C до 70°C
Розміри плати	49 × 26 мм	51 × 28 мм	51 × 21 мм	18,5 мм × 43,2 мм
Ціна	2,5\$	5\$	6\$	15\$

Після аналізу всіх згаданих платформ було вирішено обрати для виконання кваліфікаційної роботи LoLin NodeMCU v3 ESP8266, оскільки вона виділяється перш за все своєю низькою ціною, достатньою продуктивністю для виконання поставлених завдань, невеликим розміром, наявністю необхідних бездротових технологій для передачі інформації користувачу та доступністю на ринку.

### **3.2 Детальний опис обраної апаратної платформи та її компонентів**

Зважаючи на той факт, що бюджет дуже обмежений, в якості апаратної платформи було обрано LoLin NodeMCU v3 ESP8266. До апаратної платформи підключено 6-осьовий модуль гіроскопа з акселерометром GY-521 (MPU - 6050). За наявної можливості, можна додати ще один модуль GY-521 для отримання більш точних результатів вимірювання. Для живлення було обрано LiSOCl2 батарею на 2400 мА/год з напругою у 3,6 В якої вистачає для живлення плати та модуля. Система працює у бездротовій мережі за допомогою WiFi-маршрутизатора TP-Link WR841N.

На рис. 3.1 зображено зовнішній вигляд LoLin NodeMCU v3 ESP8266.



Рисунок 3.1 – Зовнішній вигляд плати LoLin NodeMCU v3 ESP8266

У табл. 3.2 наведено технічні характеристики плати.

Таблиця 3.2 – Технічні характеристики LoLin NodeMCU v3 ESP8266

MCU	Tensilica L106 32-біт
Частота	80 – 160 МГц
Живлення	3,3 В
Вхідна напруга	7 – 12 В
Пам'ять SRAM	64 КБ
Пам'ять FLASH	До 4 МБ
Пам'ять EEPROM	4 КБ
Бездротовий зв'язок	802.11 b/g/n
GPIO	17
UART/SPI/I2C/CAN/SDIO	1/1/1/0/0
Діапазон АЦП	0 – 3,3 В
Струм споживання	130 мА
Робочий діапазон температур	Від -40°C до 85°C
Розміри плати	49 × 26 мм
Ціна	2,5\$



З технічних характеристик можна побачити, що вона досить невеликого розміру:  $49 \times 26$  мм, містить 17 портів загального призначення та має модуль бездротової мережі який може працювати за стандартом IEEE 802.11 b/g/n, що і потрібно для проектування прототипу.

На рис. 3.2 зображено розпінування плати LoLin NodeMCU v3 ESP8266.

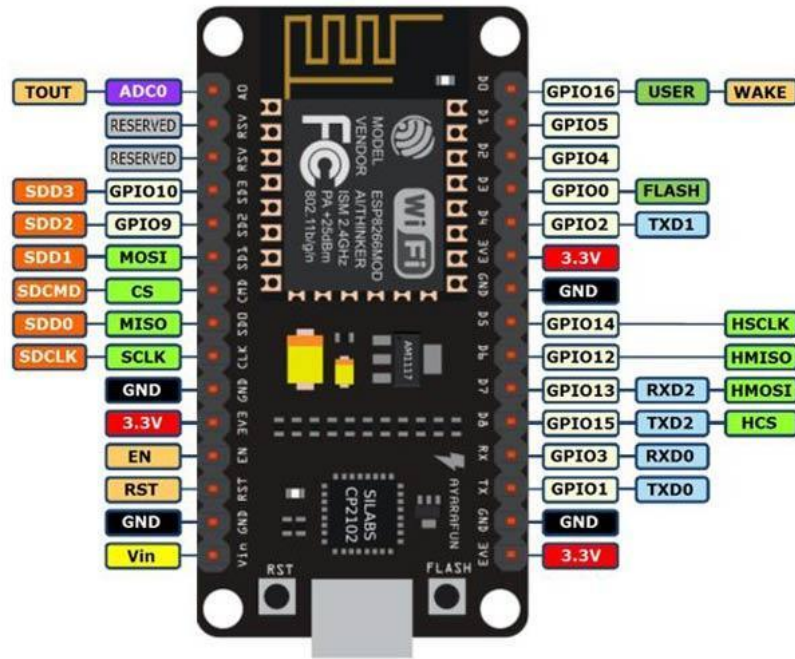


Рисунок 3.2 – Розпінування плати LoLin NodeMCU v3 ESP8266

### 3.2.1 Модуль гіроскопа та акселерометра

Для визначення положення пристрою і прискорення було обрано 6-осьовий модуль гіроскопа та акселерометра GY-521 (MPU-6050).

Модуль GY-521 – це плата для MPU-6050 MEMS, яка має 3-осьовий гіроскоп, 3-осьовий акселерометр, цифровий процесор руху і датчик температури. На рис. 3.3 зображено зовнішній вигляд модуля.



Рисунок 3.3 – Модуль GY-521

У табл. 3.3 наведено технічні характеристики модуля GY-521.

Таблиця 3.3 – Технічні характеристики модуля GY-521

Чіп	MPU-6050
Живлення	Від 3,3 до 5 В
Діапазон вимірювання гіроскопа	$\pm 250$ °/с, $\pm 500$ °/с, $\pm 1000$ °/с, $\pm 2000$ °/с
Діапазон вимірювання акселерометра	$\pm 2g$ , $\pm 4g$ , $\pm 8g$ , $\pm 16g$
Інтерфейс підключення	I2C
Розміри	21 мм × 16 мм × 3 мм
Вага	5 г

### 3.2.2 Живлення

В якості живлення було обрано літєву батарейку Li-SOCL2 [25]. Елементи живлення електрохімічної схеми Li-SOCL2 гарантують довгий термін використання при температурах від  $-55^{\circ}$  до  $85^{\circ}$ .

На рис. 3.4 продемонстровано зовнішній вигляд обраного джерела живлення.



Рисунок 3.4 – Літій-тіоніл-хлорид батарейка для живлення плати

У табл. 3.4 наведено характеристики літій-тіоніл-хлоридного джерела живлення.

Таблиця 3.4 – Характеристики Li-SOCI2 джерела живлення

Ємність	2400 мА/год
Найвища робоча температура	85° C
Найнижча робоча температура	-55° C
Напруга	3,6 В

### 3.2.3 WiFi-маршрутизатор

Для надсилання повідомлень користувачу про виявлення падіння потрібно мати точку доступу до якої ESP8266 зможе під'єднатись. Для виконання цієї задачі було обрано маршрутизатор TP-Link WR841N. На рис. 3.5 показано зовнішній вигляд маршрутизатора.



Рисунок 3.5 – Маршрутизатор TP-Link WR841N

У табл. 3.5 наведено технічні характеристики маршрутизатора.

Таблиця 3.5 – Технічні характеристики TP-Link WR841N

Роз'єми	4×10/100 Мбіт/с LAN, 1×10/100 Мбіт/с WAN
Стандарти	IEEE 802.11b/g/n
Підтримувані технології	MIMO, QSS
Швидкість	До 300 Мбіт/с (802.11n)
Частоти	2,400 – 2,4835 ГГц
Потужність передавача	20 дБм
Режими WAN	Dynamic IP, Static IP, PPPoE, PPTP(Dual Access), L2TP(Dual Access), BigPond
Шифрування	WEP, WPA, WPA2, WPA/WPA2-Enterprise (802.1x)
Робоча температура	від 0°C до +40°C
Розміри	192 × 130 × 33 мм

### 3.3 Побудова принципової та макетної схеми

Перед початком конструювання прототипу, макетна та принципові схеми були спроектовані у середовищі проектування Fritzing [26].

Fritzing – це широко використовувана програмна платформа з відкритим кодом, спеціально розроблена для створення прототипів електроніки та проектування схем. Платформа пропонує комплексний та інтуїтивно зрозумілий інтерфейс, який дозволяє користувачам створювати та тестувати електронні схеми за допомогою візуального представлення.

Програмне забезпечення надає велику бібліотеку електронних компонентів, включаючи мікроконтролери, датчики, актуатори та різні інші компоненти, які можна легко розмістити на віртуальній макетній платі. З'єднання між компонентами можуть бути встановлені і перевірені, а поведінка схеми може бути змодельована з метою тестування.

Елементи які входять до базового комплексу не включають в собі плату NodeMCU v3 ESP8266 та модуль GY-521, тому їх було завантажено з офіційного веб-сайту де користувачі діляться своїми компонентами.

Після успішного встановлення компонентів, було зроблено макетну та принципові схеми прототипу (рис. 3.6).

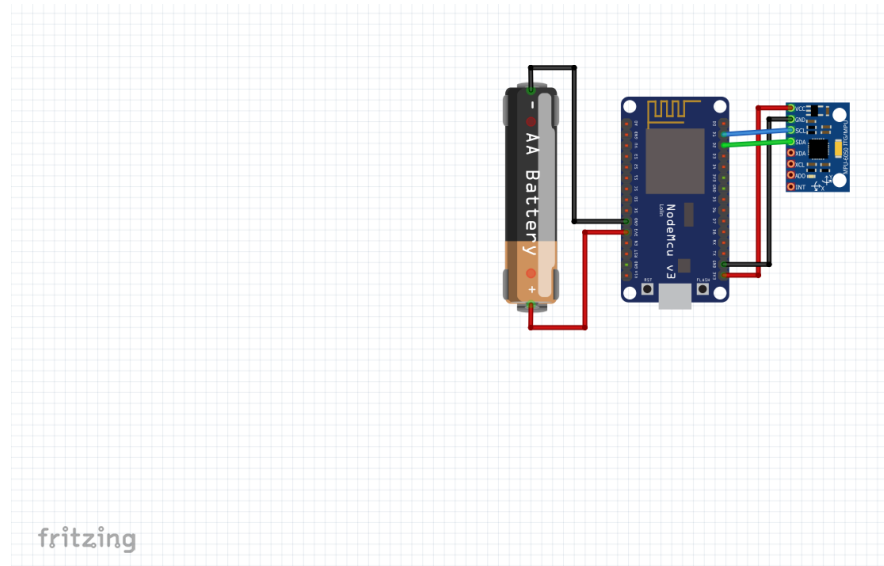


Рисунок 3.6 – Макетна схема прототипу

На рис. 3.7 зображено принципову схему прототипу.

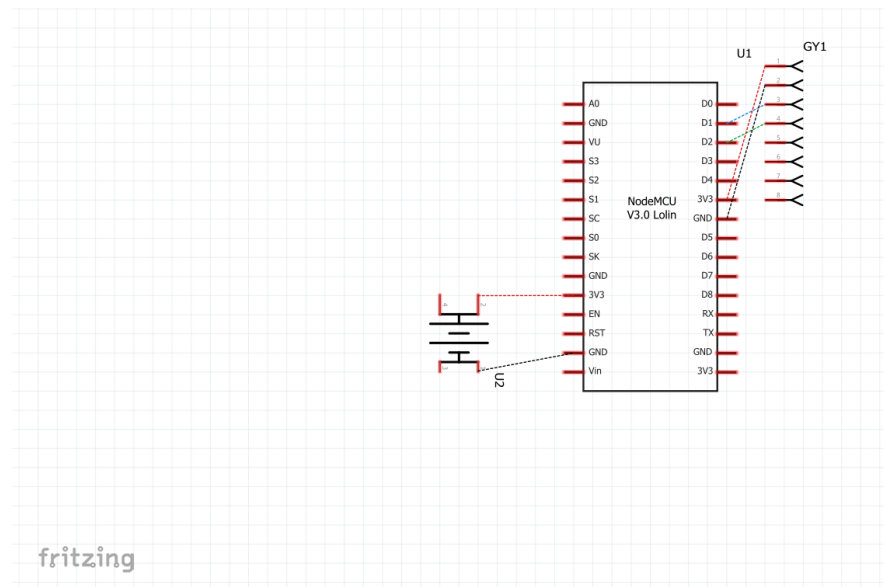


Рисунок 3.7 – Принципова схема прототипу

Також було спроектовано макетну схему для ситуації, коли використовується 2 модуля GY-521 (рис. 3.8).

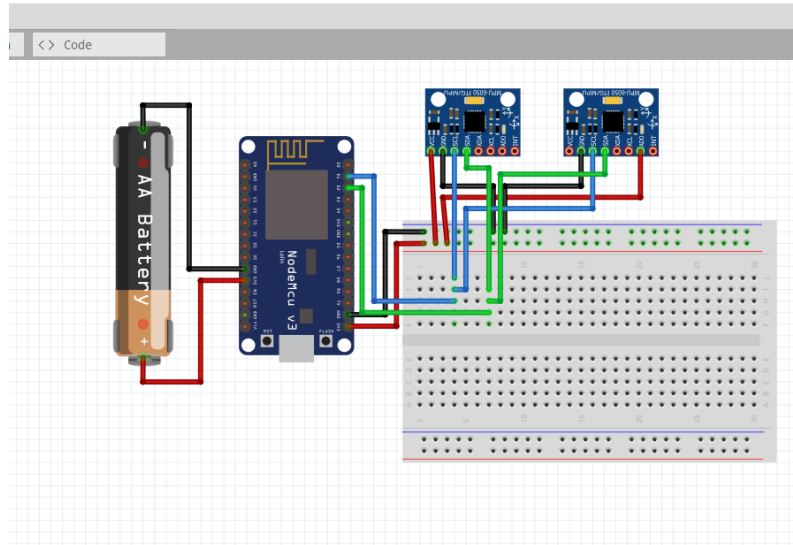


Рисунок 3.8 – Макетна схема прототипу з двома модулями GY-521

У цьому випадку пін живлення VCC першого модуля та пін AD0 другого модуля під'єднуються до живлення 3,3 В для того, щоб плата ESP8266 була здатна перемикається між I2C інтерфейсами та зчитувати дані з двох модулів.

### 3.4 З'єднання модуля GY-521 з апаратною платформою

Після того як була побудована макетна та принципова схеми, було розпочато конструювання прототипу. До плати NodeMCU ESP8266 було під'єднано модуль гіроскопа та акселерометра GY-521:

- Пін VCC модуля було з'єднано з піном 3,3 В;
- пін GND модуля з'єднано з піном GND плати;
- пін SCL модуля з'єднано з піном загального призначення D1 плати;
- пін SDA модуля з'єднано з піном загального призначення D2.

Результат продемонстровано на рис. 3.9

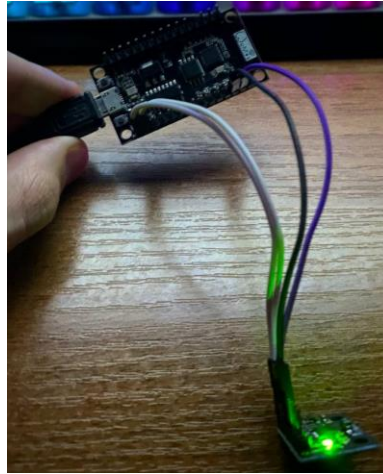


Рисунок 3.9 – Результат під'єднання модуля до плати ESP8266

### 3.5 Розробка програмної частини

Перед початком розробки програмної частини, у середовищі draw.io було розроблено алгоритм за яким потрібно щоб програмна частина працювала. На рис. 3.10 зображено алгоритм роботи програмної частини.

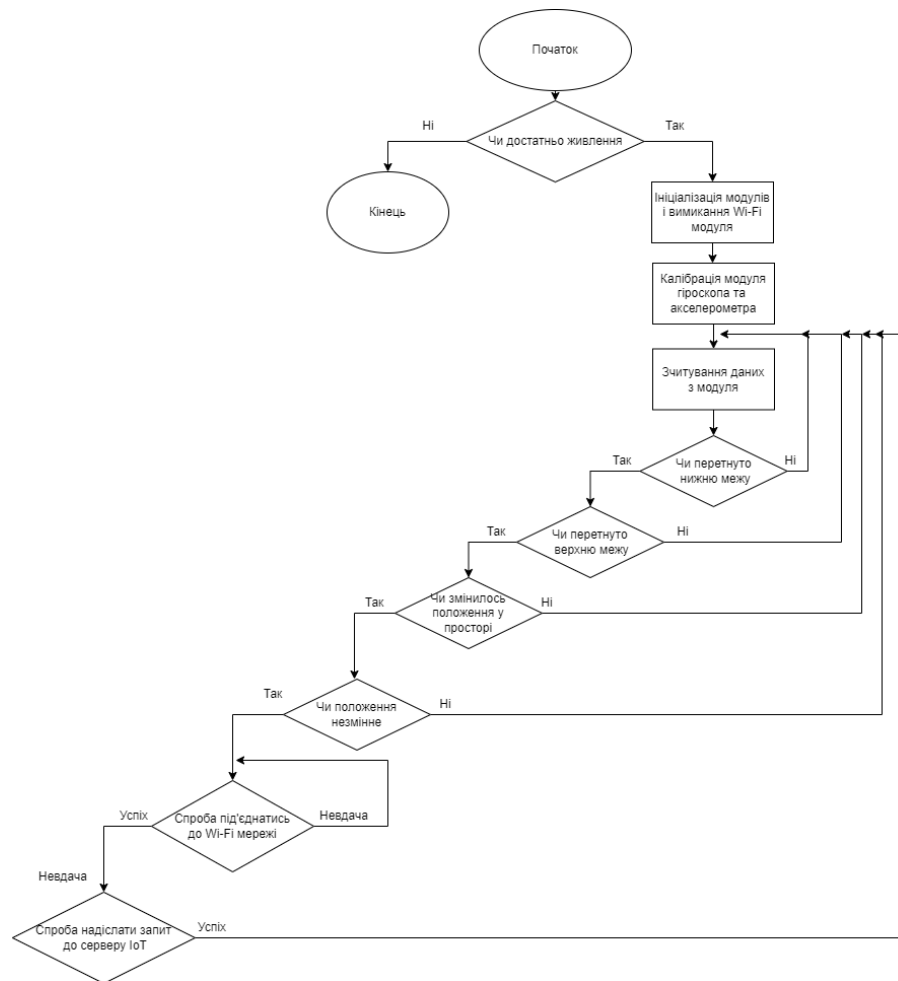


Рисунок 3.10 – Алгоритм роботи

На рис. 3.10 можна побачити, що спочатку йде перевірка чи достатньо живлення, далі йде ініціалізація модулів та під'єднання до Wi-Fi. Наступним кроком йде калібрація модуля GY-521 та зчитування з нього даних. Якщо нижня та верхня межі були перетнуті, а також положення гіроскопа змінилось, відбувається повідомлення користувача через Email та SMS.

### 3.5.1 Вибір мови програмування

Мова програмування C була обрана основною для розробки програмної частини. Використання мови програмування C для ESP8266 має ряд переваг, серед яких:

- Ефективне використання пам'яті: C є низькорівневою мовою, яка забезпечує тонкий контроль над використанням пам'яті. Це важливо при роботі з ESP8266, оскільки його обмежені ресурси пам'яті можуть легко стати вузьким місцем;
- швидкодія: C – це скомпільована мова, яка відома своєю швидкістю виконання. Програми, написані на C, можуть працювати швидше і ефективніше, ніж програми, написані на мовах вищого рівня, що робить її добре придатною для середовищ з обмеженими ресурсами, таких як ESP8266;
- підтримка сумісності: C є широко використовуваною мовою і підтримується великою кількістю інструментів розробки та бібліотек. Це дозволяє легко інтегрувати код, написаний на C, з іншими компонентами, такими як датчики, актуатори та інші мікроконтролери;
- переносимість: програми, написані на C, можна легко перенести на інші платформи та пристрої. Це означає, що код, написаний для ESP8266, можна повторно використовувати на інших мікроконтролерах, скорочуючи час і зусилля, необхідні для розробки нових проектів.



### 3.5.2 Вибір середовища розробки

В якості середовища розробки було використано середовище Arduino IDE.

Arduino IDE [27] слугує комплексною платформою для кодування та програмування плат, що має крос-платформну функціональність. Використовуючи мову програмування C/C++, воно полегшує написання коду завдяки таким функціям, як підсвічування синтаксису, автозавершення коду, виявлення помилок та підсвічування коду для оптимізації процесу розробки.

Основним компонентом Arduino IDE є редактор коду, який надає зручний інтерфейс для створення та модифікації скетчів. Редактор коду оснащений функціями, які покращують читабельність та налагоджуваність коду, включаючи підсвічування синтаксису, автоматичний відступ та виявлення помилок.

Послідовний монітор є незамінним інструментом для зв'язку між платою та програмним забезпеченням. Він дозволяє надсилати та отримувати дані, реєструвати послідовні дані та налагоджувати, забезпечуючи важливий шлях для обміну даними.

Менеджер бібліотек є невід'ємною частиною Arduino IDE, що дозволяє легко інтегрувати додаткові функції в скетчі. Він надає доступ до великої бібліотеки заздалегідь написаних кодів, так званих бібліотек, спрямованих на спрощення загальних завдань, таких як збір даних з датчиків, керування двигуном або виведення даних на РК-дисплей.

На рис. 3.11 продемонстровано зовнішній вигляд Arduino IDE.

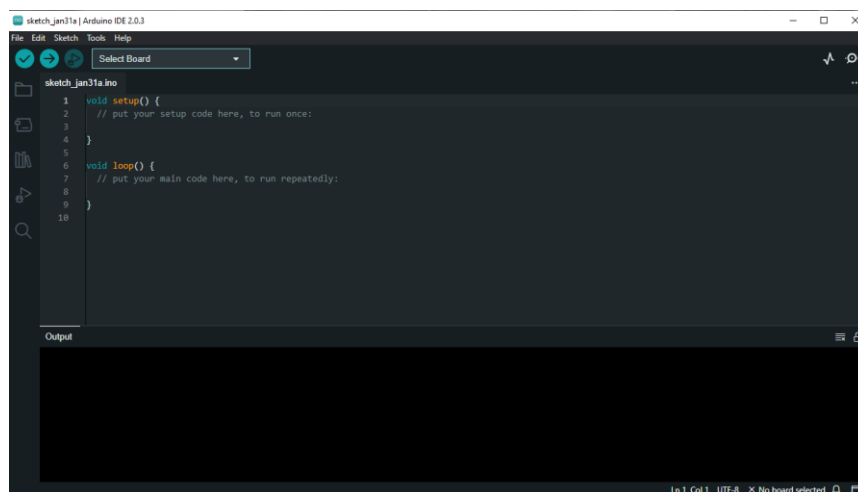


Рисунок 3.11 – Середовище розробки Arduino IDE

### 3.5.3 Бібліотеки

Використання бібліотек в інтегрованому середовищі розробки Arduino (IDE) дозволяє розширити можливості платформи та спростити процес розробки для користувачів. Бібліотеки складаються з попередньо написаних модулів коду, які полегшують виконання різних функцій, таких як зчитування даних з датчика, керування виконавчим механізмом або встановлення мережевого з'єднання.

Доступ до цих бібліотек надається через вбудований менеджер бібліотек Arduino IDE, що полегшує інтеграцію в проекти користувачів. Бібліотеки систематично класифіковані, що дозволяє розробникам легко знайти та використати найбільш підходящу бібліотеку для конкретних вимог проекту.

Включення бібліотеки до проекту передбачає включення її заголовного файлу та виклик її функцій. У заголовному файлі описується використання функцій, запропонованих бібліотекою, що дозволяє легко інтегрувати їх в ескіз проекту для виконання конкретних завдань.

Для розробки програмної частини були використані бібліотеки Wire для комунікації з I2C інтерфейсом та ESP8266WiFi для роботи з Wi-Fi чіпом плати.

### 3.5.4 Бібліотека Wire

Бібліотека Wire [28], невід'ємний компонент інтегрованого середовища розробки Arduino (IDE), є інструментом, який спрощує зв'язок між пристроями, використовуючи протокол I2C (Inter-Integrated Circuit). Ця бібліотека спрощує складнощі, притаманні комунікації I2C, і розширює її доступність для ширшого кола користувачів.

Бібліотека Wire надає зручний інтерфейс для передачі та отримання даних через протокол I2C. Протокол I2C, з його можливістю підключення декількох ведучих і ведених пристроїв, є широко розповсюдженим методом зв'язку між мікроконтролерами та іншими пристроями.

Включення бібліотеки Wire в проект є простим процесом. Він вимагає включення заголовного файлу бібліотеки та реалізації її функцій у схемі проекту. Бібліотека Wire пропонує такі функції, як запуск і завершення передачі, запис і зчитування даних, а також керування швидкістю та адресацією шини I2C.

### 3.5.5 Бібліотека ESP8266WiFi

Бібліотека ESP8266WiFi [29] є важливим інструментом в інтегрованому середовищі розробки (IDE) Arduino, спеціально розробленим для мікроконтролера ESP8266. Ця бібліотека містить повний набір функцій, що забезпечує безперешкодну взаємодію з інтегрованими можливостями Wi-Fi ESP8266.

Завдяки використанню цієї бібліотеки користувачі можуть легко підключатися до мереж Wi-Fi, встановлювати точку доступу Wi-Fi, передавати і приймати дані по мережі, а також легко адмініструвати мережеві конфігурації. Крім того, бібліотека може похвалитися можливістю безпечного Wi-Fi зв'язку, що забезпечується завдяки використанню протоколів шифрування SSL/TLS.

Інтеграція бібліотеки ESP8266WiFi у ваш проект - це безпроблемний процес, який вимагає включення заголовного файлу бібліотеки з подальшим використанням її функцій в ескізі проекту. Інтерфейс бібліотеки розроблений таким чином, щоб бути зрозумілим і зручним для користувача, що дозволяє користувачам інтегрувати функції Wi-Fi в свої проекти ESP8266 з надзвичайною легкістю.

### 3.5.6 Використанні змінні та функції

#### 1. Завантаження необхідних модулів.

```
#include <Wire.h> //бібліотека Wire
#include <math.h> // Бібліотека math для математичних обчислень
#include <ESP8266WiFi.h> //бібліотека ESP8266WiFi
```

#### 2. Ініціалізація змінних.

```
#define THRESHOLD_LOW 2.5 // Нижня межа
#define THRESHOLD_HIGH 3.5 // Верхня межа
const int addr=0x68; // I2C адреса GY-521
int16_t AccelX,AccelY,AccelZ,TempI,GyroX,GyroY,GyroZ; // Змінні для зберігання
даних з GY-521
int16_t accel_offset_x, accel_offset_y, accel_offset_z; // Дані калібровки
акселерометра
int16_t gyro_offset_x, gyro_offset_y, gyro_offset_z; // Дані калібровки
гіроскопа
float accx=0, accy=0, accz=0, gyrx=0, gyry=0, gyrz=0;
boolean fallDetected = false; // булеве значення якщо було виявлено падіння
boolean tr_lower=false; // булеве значення якщо була перетнута нижня межа
boolean tr_higher=false; // булеве значення якщо була перетнута верхня межа
boolean tr_gyroAngle=false; // булеве значення якщо було змінено положення
гіроскопа
```

#### 3. Запуск платформи, перший запуск модулів та вимикання Wi-Fi для зменшення споживання.

```
WiFi.mode( WIFI_OFF ); // Вимикаємо Wi-Fi модуль для зменшення споживання
WiFi.forceSleepBegin();
delay( 1 );
WiFi.persistent( false ); // Вимикаємо завантаження налаштувань з FLASH
пам'яті
Serial.begin(115200);
```

```
Wire.begin();  
Wire.beginTransmission(addr); // Початок комунікації з I2C до якого  
під'єднаний модуль гіроскопа та акселерометра  
Wire.write(0x6B); // PWR_MGMT_1 register  
Wire.write(0); // Ініціалізація модуля GY-521  
Wire.endTransmission(true);
```

#### 4. Функція зчитування даних з GY-521.

```
Wire.beginTransmission(addr);  
Wire.write(0x3B); // Виклик реєстра 0x3B  
Wire.endTransmission(false);  
Wire.requestFrom(addr,14,true); // Запит на отримання інформації з 14  
регістрів  
AccelX=Wire.read()<<8|Wire.read();  
AccelY=Wire.read()<<8|Wire.read();  
AccelZ=Wire.read()<<8|Wire.read();  
TempI=Wire.read()<<8|Wire.read();  
GyroX=Wire.read()<<8|Wire.read();  
GyroY=Wire.read()<<8|Wire.read();  
GyroZ=Wire.read()<<8|Wire.read();
```

#### 5. Функція надсилання повідомлення користувачу.

```
WiFiClient client; // Клас для створення TCP з'єднання  
const int Port = 80;  
String URL = "/trigger/" + event + "/with/key/" + userInfo  
// Робимо GET запит до If This Then That сервісу  
client.print(String("GET ") + url + " HTTP/1.1\r\n" + "Host: " + host + "\r\n"  
+ "Connection: close\r\n\r\n");  
client.stop(); // Завершуємо з'єднання після запиту
```

#### 6. Функція первісної калібрації модуля.

```
for(int i=0;i<1000;i++){  
Wire.beginTransmission(addr);  
Wire.write(0x3B); // Виклик реєстра 0x3B  
Wire.endTransmission(false);  
Wire.requestFrom(addr,14,true); // Запит на отримання інформації з 14  
регістрів  
AccelX=Wire.read()<<8|Wire.read();  
AccelY=Wire.read()<<8|Wire.read();  
AccelZ=Wire.read()<<8|Wire.read();  
TempI=Wire.read()<<8|Wire.read();  
GyroX=Wire.read()<<8|Wire.read();
```

```
GyroY=Wire.read()<<8|Wire.read();
GyroZ=Wire.read()<<8|Wire.read();
accel_offset_x += AcX;
accel_offset_y += AcY;
accel_offset_z += AcZ;
gyro_offset_x += GyX;
gyro_offset_y += GyY;
gyro_offset_z += GyZ;
delay(2);
}
accel_offset_x /= 1000;
accel_offset_y /= 1000;
accel_offset_z /= 1000;
gyro_offset_x /= 1000;
gyro_offset_y /= 1000;
gyro_offset_z /= 1000;
```

## 7. Функція перевірки отриманих даних на перетинання вказаних меж

```
// Обчислюємо величину прискорення
float acc_magnitude = sqrt(accx * accx + accy * accy + accz * accz);
if (acc_magnitude <= THRESHOLD_LOW && tr_higher ==false){
// Якщо перетинається нижня межа, спрацьовує перший тригер
tr_lower=true;
} if (tr_lower=true){
if (acc_magnitude >= THRESHOLD_HIGH){
// Якщо перетинається нижня межа, спрацьовує другий тригер
tr_higher=true;
}
} if (tr_higher==true){
gyro_angle = sqrt(gyrx * gyrx + gyry * gyry + gyrz * gyrz);
if (gyro_angle >=350 && gyro_angle <=440){
// Умова при якій кутова швидкість була у заданих межах
tr_gyroAngle=true;
}
} if (tr_gyroAngle ==true){
gyro_angle = sqrt(gyrx * gyrx + gyry * gyry + gyrz * gyrz);
if ((gyro_angle >=0) && (gyro_angle <=10)){
// Якщо положення гіроскопа незмінне, встановлюється флаг що користувач впав
fallDetected=true; tr_gyroAngle=false;
}} if (fallDetected ==true){
WiFi.forceSleepWake(); // Вмикаємо Wi-Fi
```

```
delay( 1 );  
WiFi.mode( WIFI_STA );  
WiFi.begin( ssid, pass );  
while (WiFi.status() != WL_CONNECTED) // З'єднуємось поки з'єднання не буде  
успішним  
{  
    delay(500);  
}  
send_event("fall_detect_email"); // Повідомлення по пошті  
send_event("fall_detect_sms"); // Повідомлення по SMS  
fallDetected=false;  
}
```

### 3.6 Діаграма послідовності роботи застосунку

На рис. 3.12 можна побачити фінальну діаграму послідовності роботи.

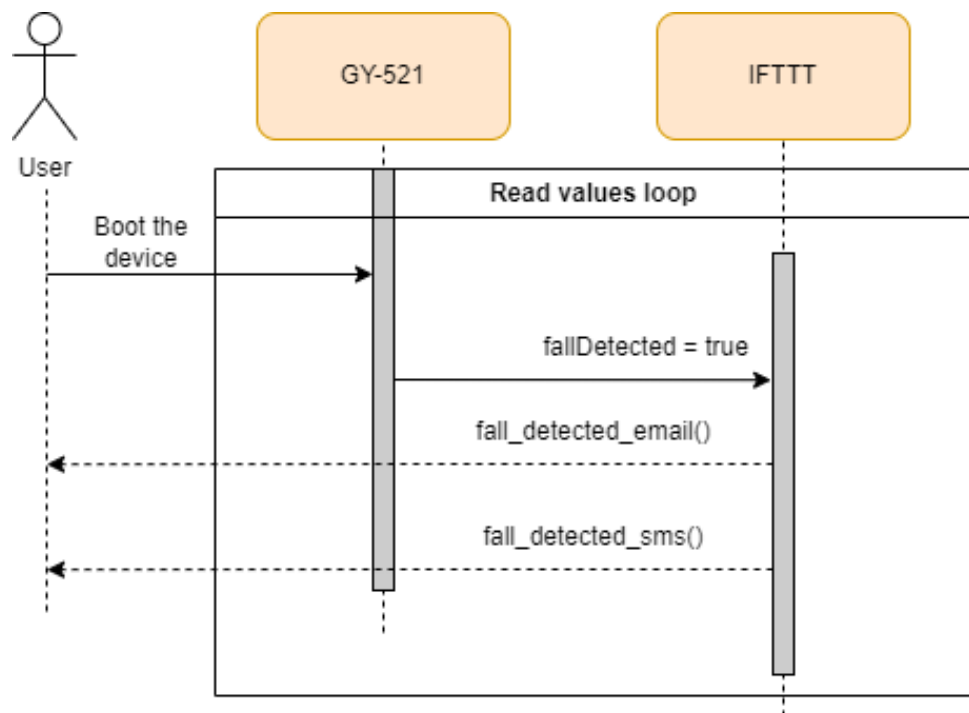


Рисунок 3.12 – Діаграма послідовності

### **Висновки до розділу 3**

На початку третього розділу наведено порівняльну таблицю можливих для використання апаратних платформ та встановлено, що NodeMCU v3 ESP8266 є найкращим вибором серед запропонованих ESP32, Raspberry Pi Pico W та Arduino Nano. Далі було описано характеристики обраної платформи та модулів які були використані під час проектування та конструювання прототипу. Програмний додаток було розроблено на мові програмування C у середовищі розробки Arduino Nano використовуючи бібліотеки для керування I2C інтерфейсом та Wi-Fi можливостями відповідно.

Згідно з розробленого алгоритму, програма повинна проводити первинну калібрацію та потім зчитувати дані з модуля гіроскопа та акселерометра GY-521 виконуючи перетворення дані з акселерометра у одиниці вимірювання еквіваленту сили тяжіння та дані з гіроскопа у градуси на секунду.



## 4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ. АНАЛІЗ РЕЗУЛЬТАТІВ ОБЧИСЛЕНЬ

### 4.1 Виконання налаштувань

Перед тим як завантажити програмну частину до апаратної платформи, потрібно провести певні налаштування, а саме:

- у налаштуваннях маршрутизатора, знайти встановлений SSID та пароль від мережі до якої буде з'єднуватись апаратна платформа;
- у сервісі IFTTT [30] інтернету речей потрібно створити дві ситуації при запиті до яких, буде надсилатись повідомлення користувачу по Email та SMS.

#### 4.1.1 Налаштування маршрутизатора TP-Link WR841N

Щоб зайти у налаштування маршрутизатора, потрібно у вікні веб-браузера перейти по наступній локальній адресі: 192.168.0.1 та ввести ім'я та пароль від акаунту адміністратора (рис. 4.1). Дана інформація зазвичай вказана позаду маршрутизатора.

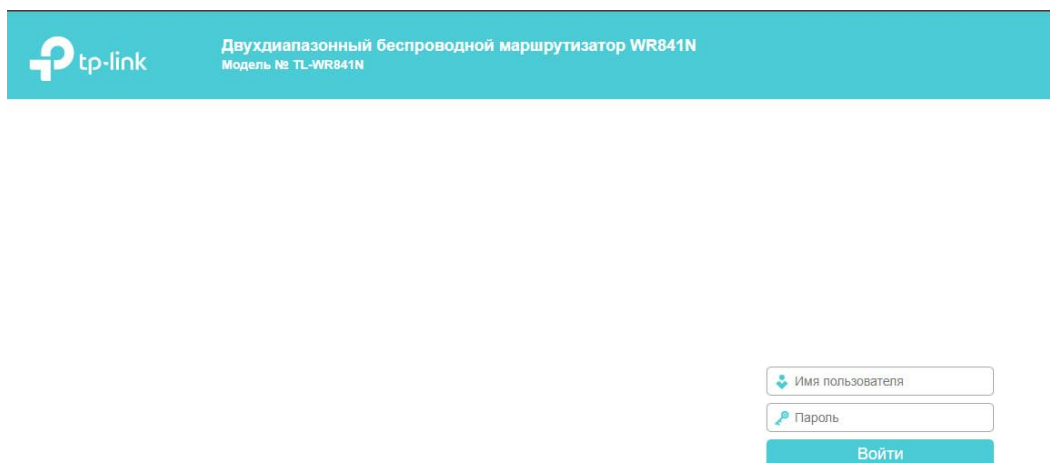


Рисунок 4.1 – Сторінка авторизації

Після успішної авторизації, у вкладці «Бездротовий режим» – «Основні налаштування» можна знайти SSID, або ім'я бездротової мережі. У моєму випадку це voloshchuk-serhii (рис. 4.2).

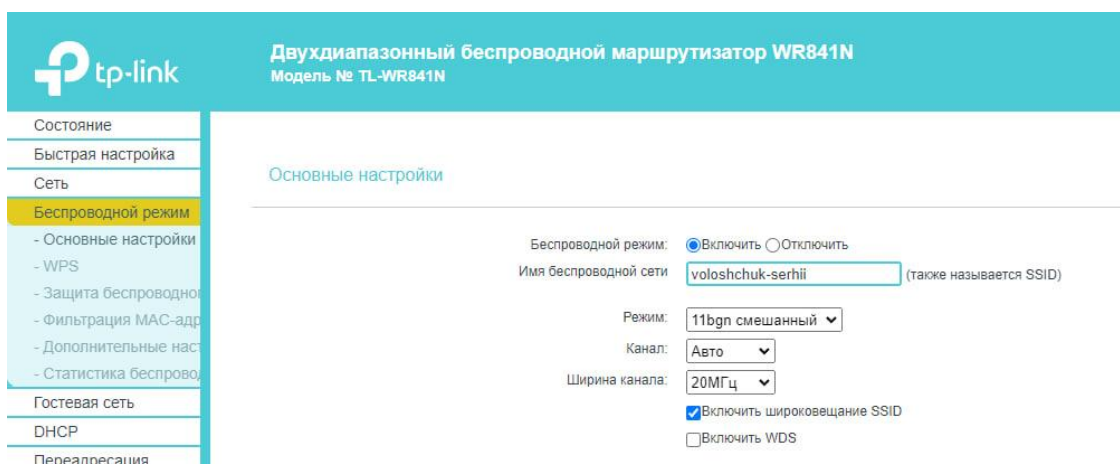


Рисунок 4.2 – Основні налаштування бездротового режиму

У вкладці «Захист бездротової мережі» можна знайти пароль від мережі. У моєму випадку це також voloshchuk-serhii (рис. 4.3).

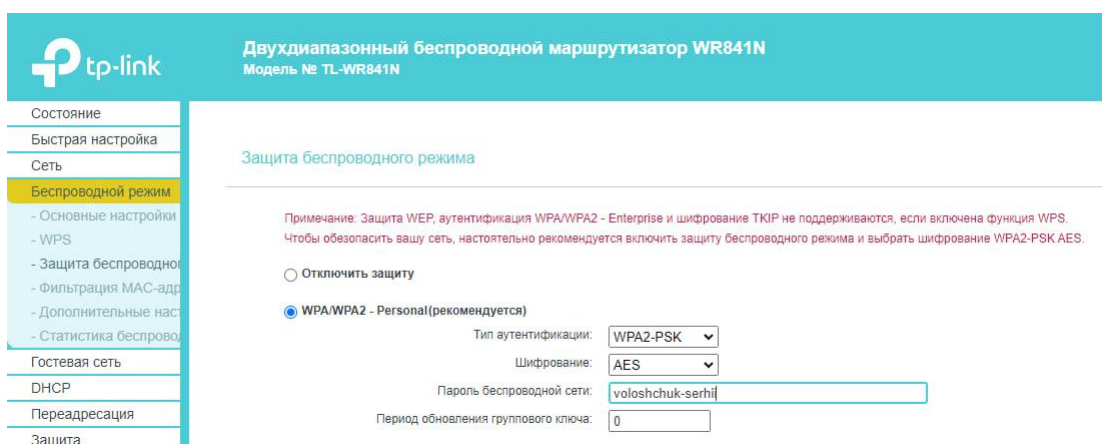


Рисунок 4.3 – Сторінка захисту бездротової мережі

Після того як потрібні дані були знайдені, у програмній частині були внесені зміни до наступних рядків:

```
const char *ssid = "voloshchuk-serhii"; // Ім'я Wi-Fi мережі  
const char *pass = "voloshchuk-serhii"; // Пароль від мережі Wi-Fi
```

#### 4.1.2 Налаштування сервісу інтернет речей IFTTT

IFTTT (If This Then That) – це веб-платформа для автоматизації, яка дозволяє користувачам створювати прості умовні твердження, відомі як "рецепти". Ці рецепти об'єднують різні веб-сервіси і дозволяють користувачам автоматизувати прості завдання та робочі процеси. Основна структура рецепту: "Якщо [це] відбувається, то відбувається [те]". [Це]

визначається як тригерна подія від одного сервісу, тоді як [те] – це дія, що виконується іншим сервісом.

Спочатку потрібно зареєструватись за адресою <https://ifttt.com/join> використовуючи пошту, або акаунти Apple, Google чи Facebook.

Після вдалої реєстрації, з'являється можливість створити нові аплети (рис. 4.4).

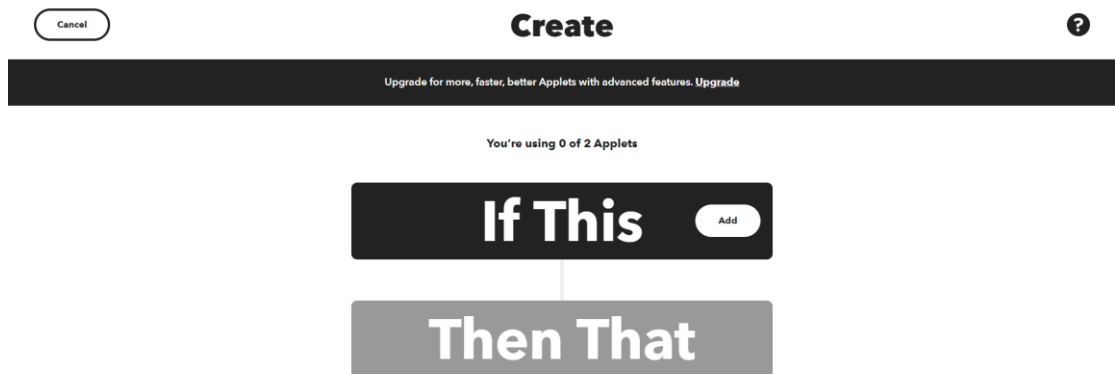


Рисунок 4.4 – Сторінка створення аплетів

Для передачі даних з апаратної платформи до сервісу IFTTT будуть використовуватись так звані веб-хуки. Тому під час створення нового тригера, потрібно обрати у вікні сервісів веб-хук (рис. 4.5)

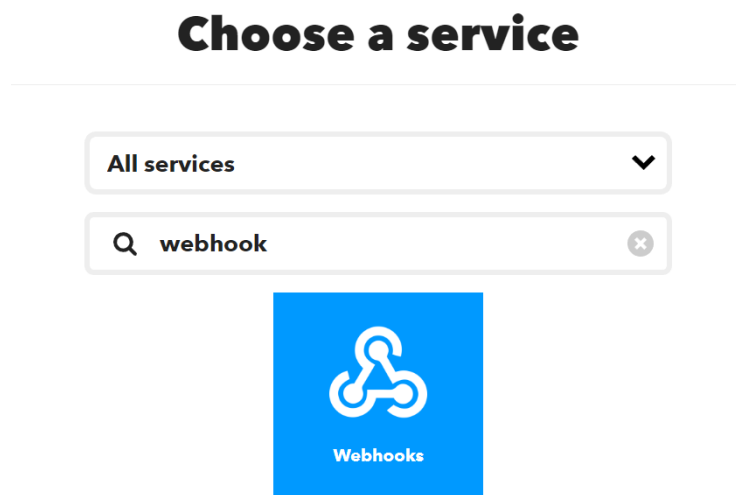


Рисунок 4.5 – Вікно вибору сервісів для тригера

Після вибору сервісу, нам запропонують обрати тригер під час отримання веб запиту. Після погодження, з'являється можливість вказати ім'я події (рис. 4.6).

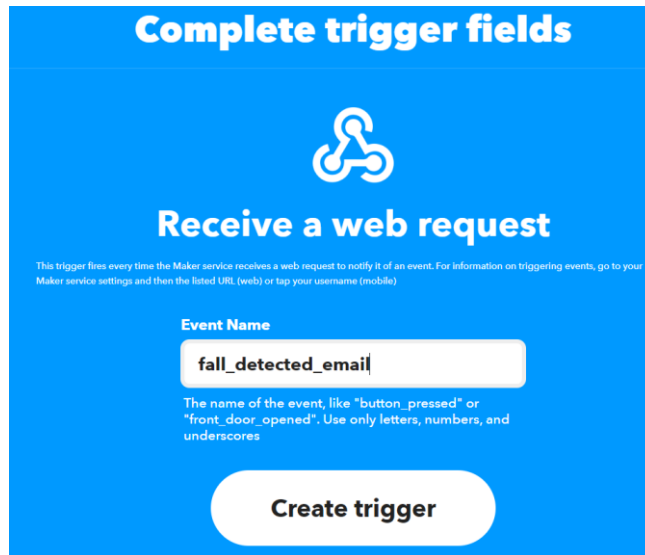


Рисунок 4.6 – Сторінка вибору ім'я події

Далі, таким самим чином обираємо сервіс для другої умови «Then That», але замість веб-хуку, буде обрано Email (рис. 4.7).

## Choose a service

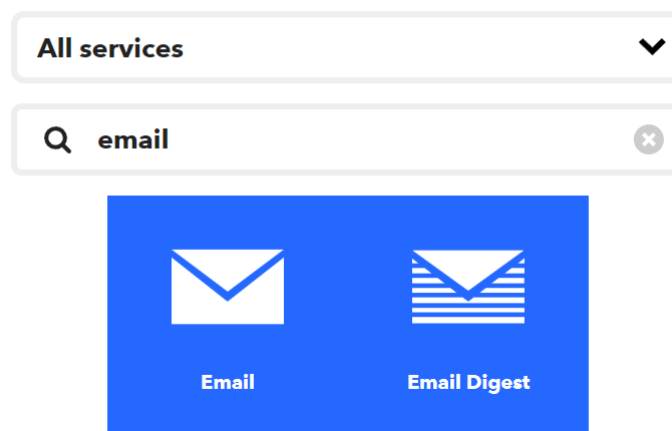


Рисунок 4.7– Вікно вибору сервісів

На наступній сторінці потрібно вказати тему Email повідомлення та тіло повідомлення. В моєму випадку темою повідомлення стане: Виявлено падіння користувача {{OccurredAt}}, де {{OccuredAt}} це дата коли подія відбулось. Тілом повідомлення стане: Користувач впав!<br>Час виявлення падіння: {{OccurredAt}}<br>Негайно зателефонуйте користувачу та дізнайтесь про його стан!<br>, де <br> це тег переносу тексту на новий рядок (рис. 4.8).

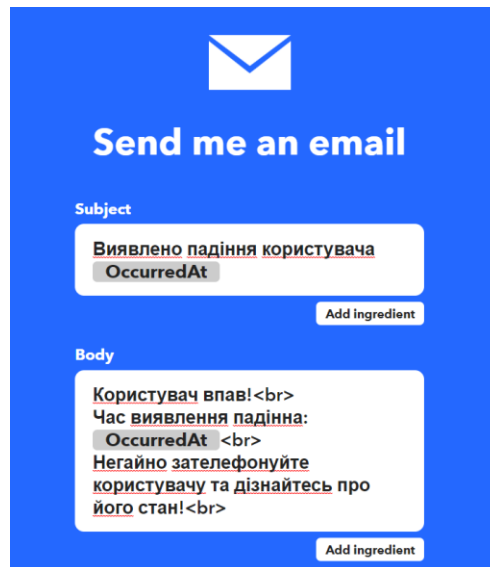


Рисунок 4.8 – Створення Email повідомлення

Для того, щоб відправити повідомлення користувачу по SMS, потрібно завантажити офіційний застосунок на мобільний пристрій та за таким же принципом створити новий аплет з назвою `fall_detected_sms` на акаунті.

На рис. 4.9 продемонстровано інтерфейс створення SMS повідомлення на мобільному телефоні Apple iPhone 11.

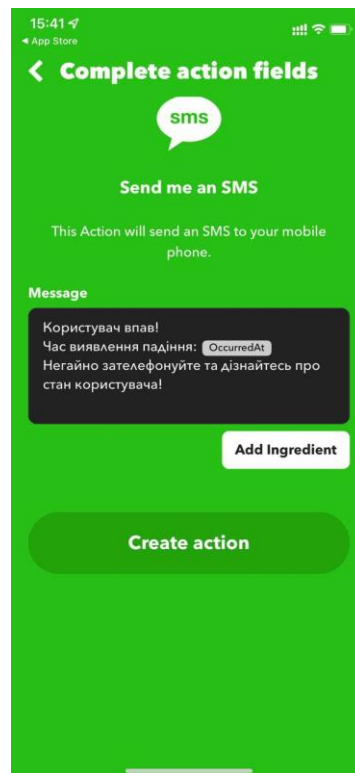


Рисунок 4.9 – Створення SMS повідомлення

Після успішного налаштування сервісу, програмну частину можна завантажувати до апаратної платформи.

## 4.2 Визначення нижньої та верхньої межі прискорення та меж кутової швидкості

Для того, щоб визначити нижню та верхню межу прискорення при падінні, було зроблено декілька експериментів.

### 4.2.1 Визначення середнього значення величини прискорення під час ходьби

Спочатку було перевірено, яке прискорення показує модуль під час звичайної ходьби. На рис. 4.10 можна побачити графік величини прискорення.

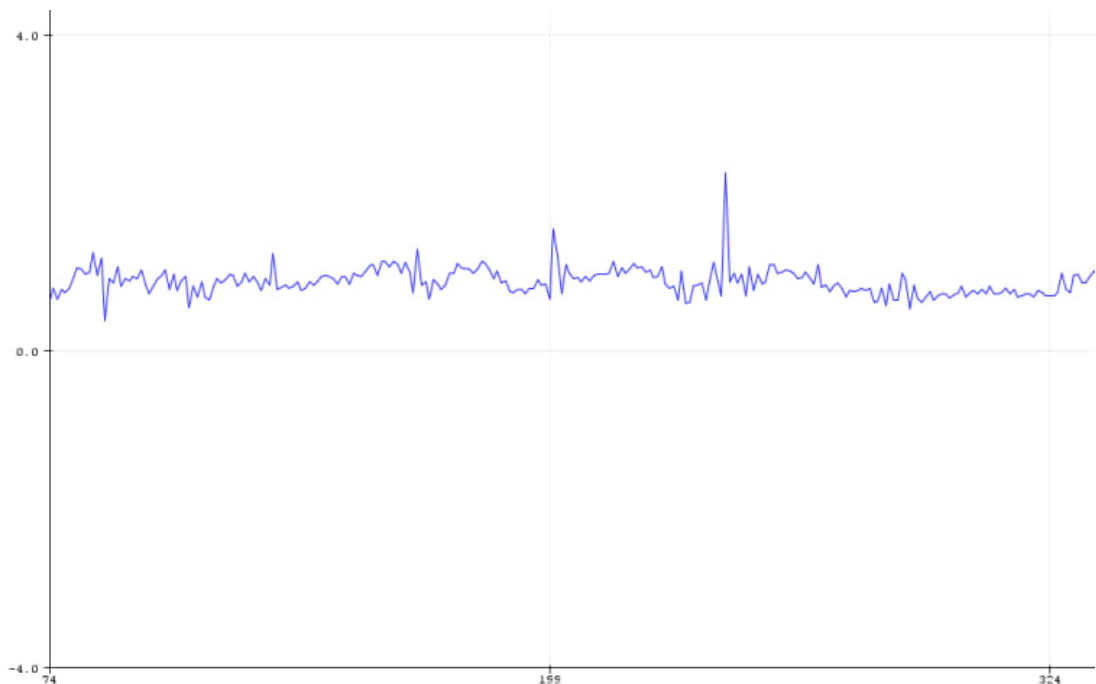


Рисунок 4.10 – Графік величини прискорення під час звичайної ходьби

З рис. 4.10 випливає, що в середньому, значення величини прискорення під час ходьби дорівнює 1,3 g.

Після цього було ще декілька раз перевірене значення величини прискорення для того, щоб переконатись у правильності вимірювання. На

рис. 4.11 та рис. 4.12 можна побачити результати вимірювань у вигляді графіків.

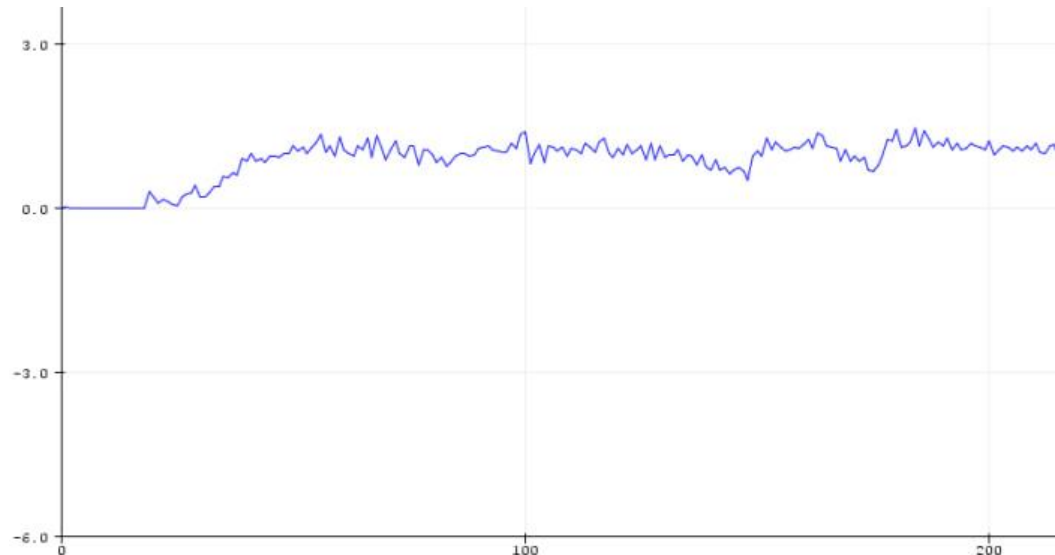


Рисунок 4.11 – Другий графік величини прискорення під час ходьби

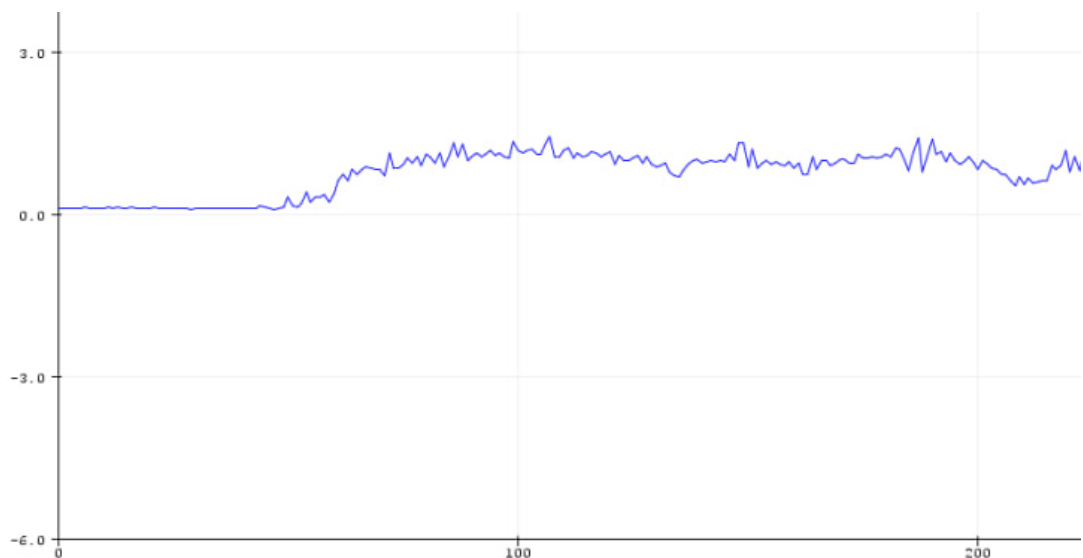


Рисунок 4.12 – Третій графік величини прискорення під час ходьби

Порівнявши графіки, можна дійти висновку що середнє значення і справді дорівнює 1,3 g.

#### 4.2.2 Визначення верхньої та нижньої межі

Для того, щоб визначити значення верхньої межі величини прискорення, було зроблено імітацію падіння, щоб побачити, яке значення

дає акселерометр з модуля. На рис. 4.13 можна побачити графік після імітації падіння.

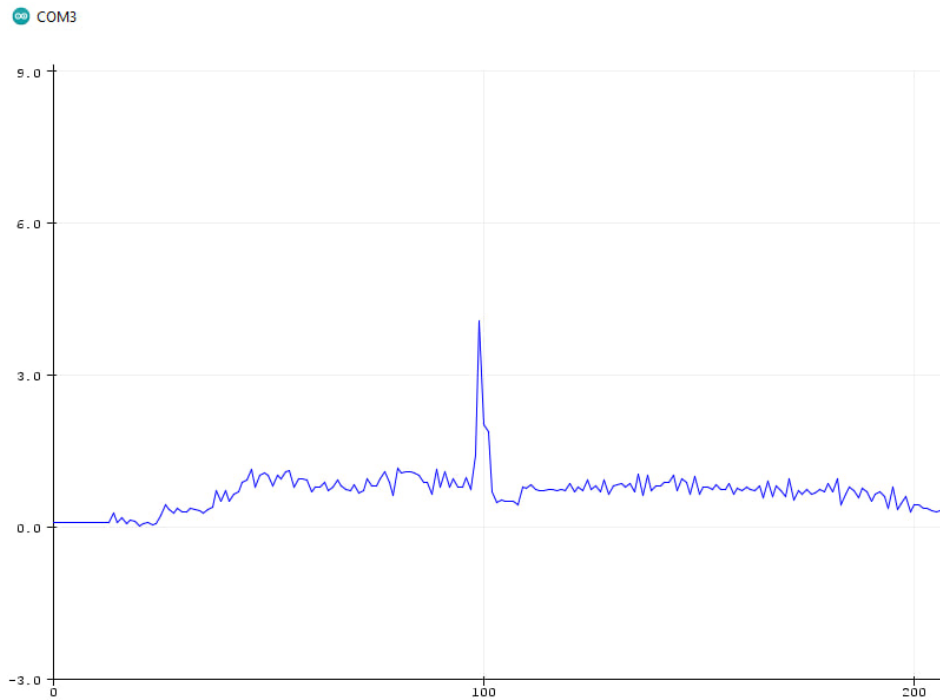


Рисунок 4.13 – Графік після імітації падіння

З рис. 4.13 видно, що зімітувавши падіння один раз, значення величини прискорення піднялось до 4,2 g. Щоб встановити кінцеве значення верхньої межі, було проведено ще декілька експериментів, графіки яких показано на рис. 4.14 та рис. 4.15.

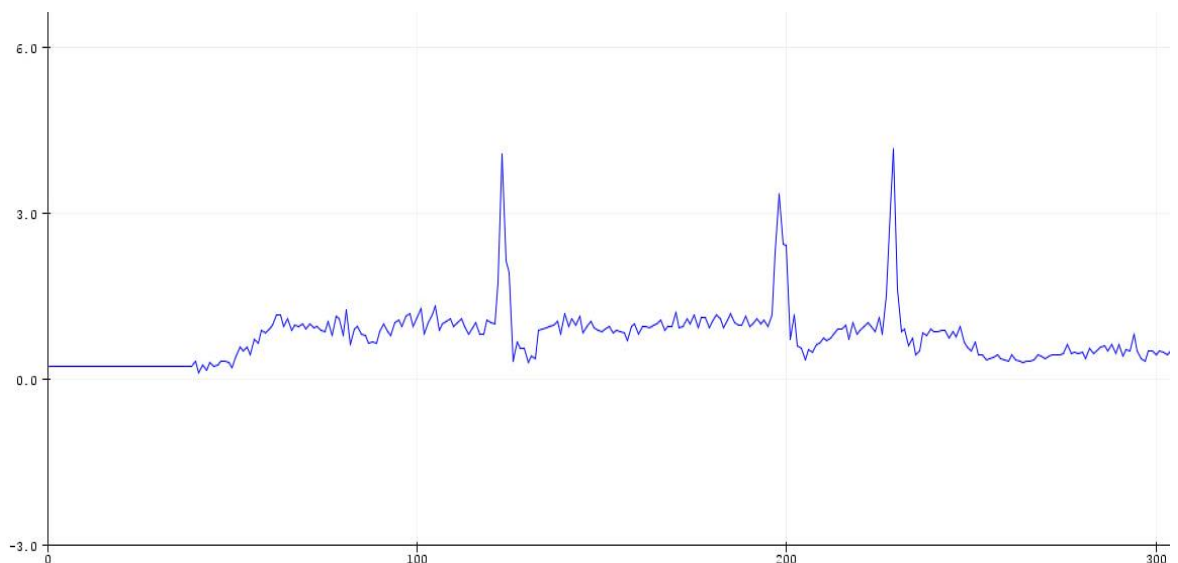


Рисунок 4.14 – Другий графік величини прискорення після імітації декількох падінь



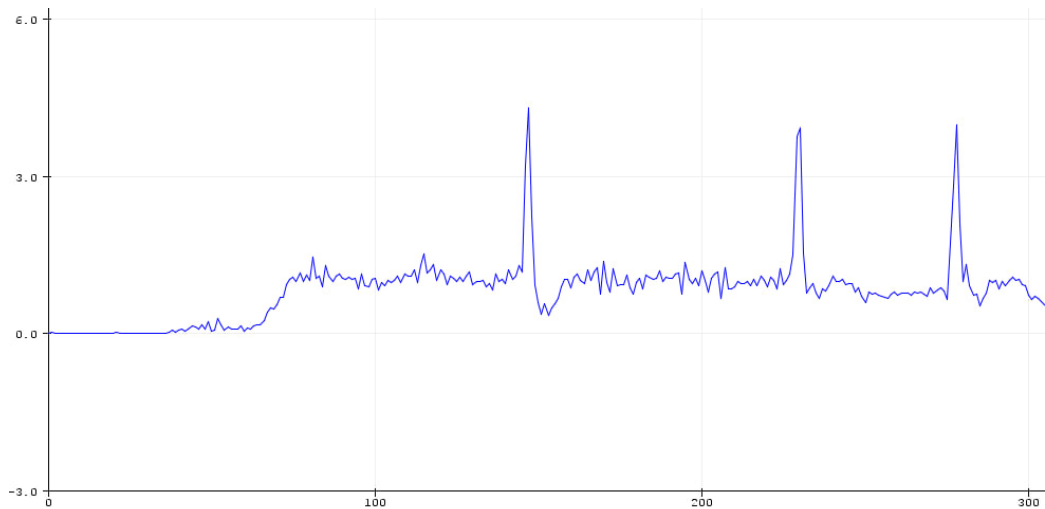


Рисунок 4.15 – Третій графік величини прискорення після імітації декількох падінь

З рис. 4.14 зрозуміло, що мінімальне значення величини прискорення при якому було імітоване падіння дорівнює 3,5 g, саме тому це значення було обране за кінцеве. Якщо повернутись до рис. 4.10, можна побачити пік у 2 g який виник під час звичайної ходьби. Це значення буде використано як нижня межа величини прискорення.

#### 4.2.2 Визначення меж кутової швидкості

Для визначення меж кутової швидкості, спочатку були проведені її вимірювання під час звичайної ходьби на рис. 4.16 зображено графік отриманих з модуля значень.

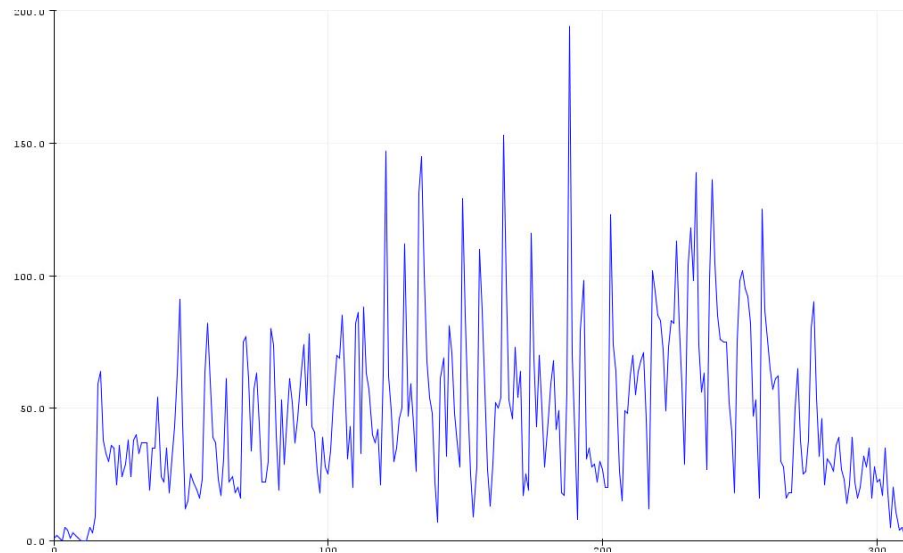


Рисунок 4.16 – Графік значень кутової швидкості під час ходьби

З рис. 4.16 випливає, що середнє значення кутової швидкості під час ходьби дорівнює 62 град/с.

Порівнявши рис. 4.16 з рис. 4.17 можна дійти висновку що значення є вірним.

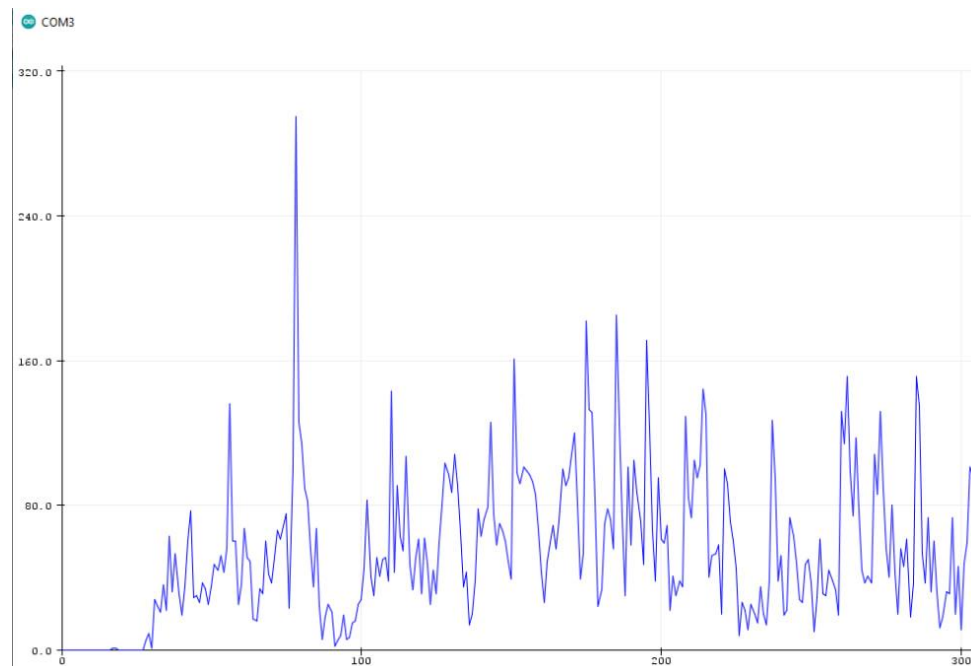


Рисунок 4.17 – Другий графік значень кутової швидкості під час ходьби

Далі було проведено декілька імітації падіння для визначення меж кутової швидкості. На рис. 4.18 зображено імітацію трьох падінь.

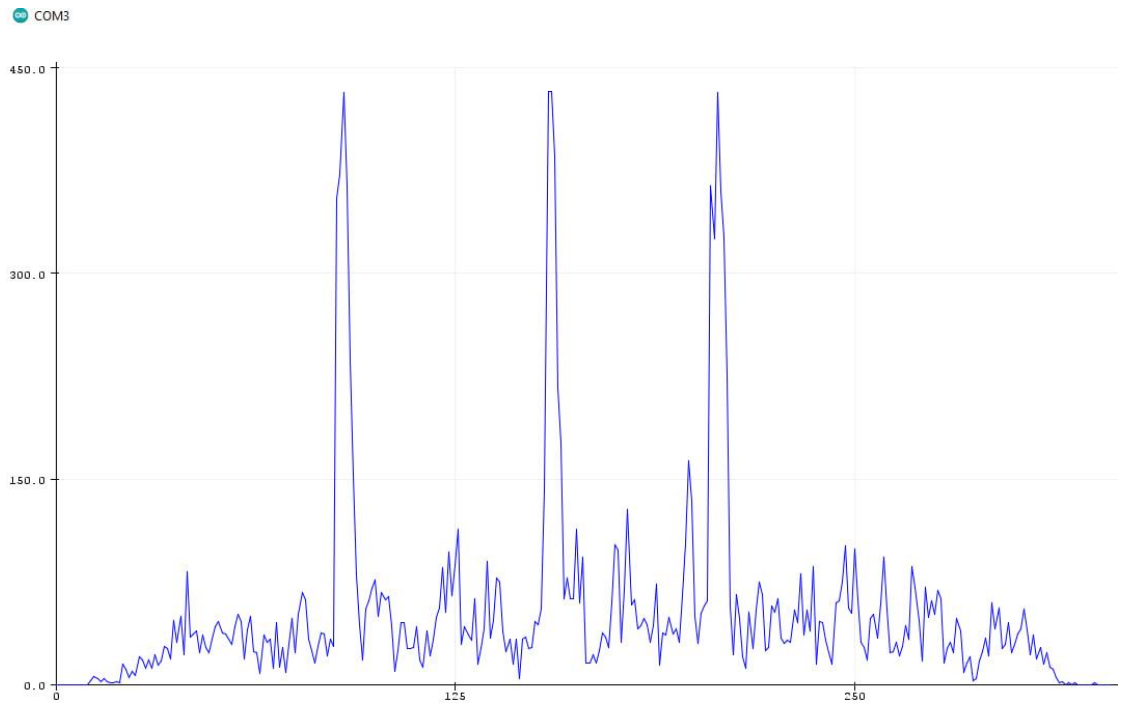


Рисунок 4.18 – Графік значень після імітації падінь

З графіку можна побачити, що під час падіння, значення кутової швидкості змінюється до значення у 440 град/с. Було проведено ще імітацію падінь для порівняння результатів (рис. 4.19).

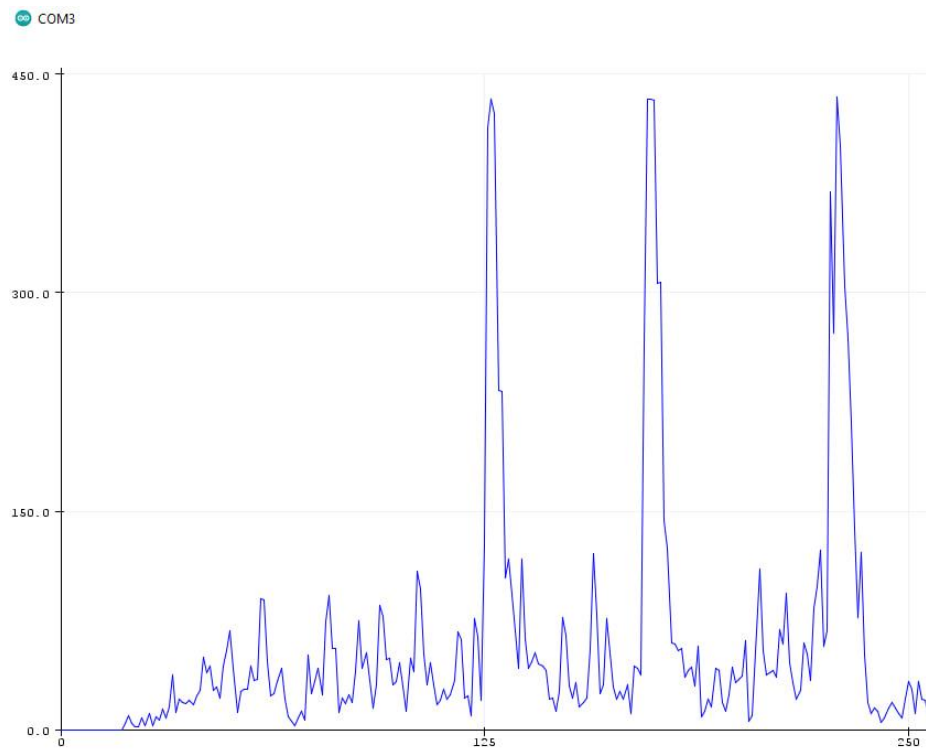


Рисунок 4.19 – Другий графік значень після імітації падінь

Порівнявши два графіки, можна дійти висновку що максимальне значення кутової швидкості під час падіння дорівнює 440 град/с. Якщо подивитись на другий графік кутової швидкості під час ходьби, можна встановити, що мінімальне значення кутової швидкості під час падіння потрібно бути вищим за 300 град/с. Тож доцільним буде встановити межі від 350 до 440 град/с враховуючи можливу похибку.

Для того, щоб встановити достовірно, що відбулось падіння, накладається додаткова умова, значення кутової швидкості після падіння має бути дуже низьким, що означає, що падіння справді трапилось (рис. 4.20).

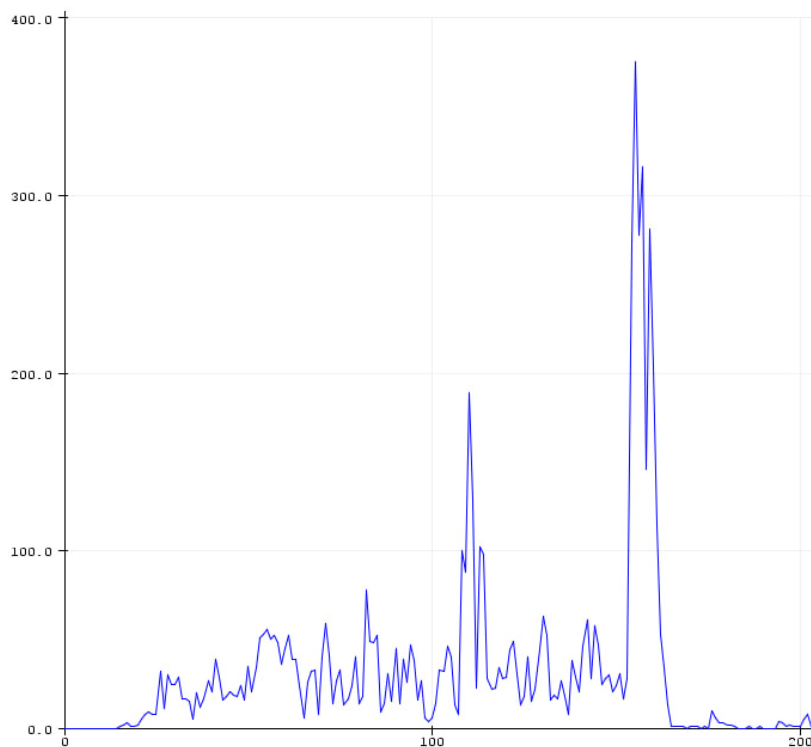


Рисунок 4.20 – Графік після імітації падіння з умовою про нерухомість

### 4.3 Перевірка працездатності

Після встановлення всіх меж у програмній частині та завантаженні оновленого коду до апаратної платформи, було перевірено працездатність всієї системи. Була проведена імітація падіння з подальшою нерухомістю платформи. Через декілька секунд, після того як відбулось падіння, платформа приєдналась до Wi-Fi мережі та надіслала 2 запити до сервісу

IFTTT. Сервіс негайно надіслав повідомлення на пошту та мобільний телефон, що відбулось падіння, що і слід було протестувати. На рис. 4.21 продемонстровано результат роботи.

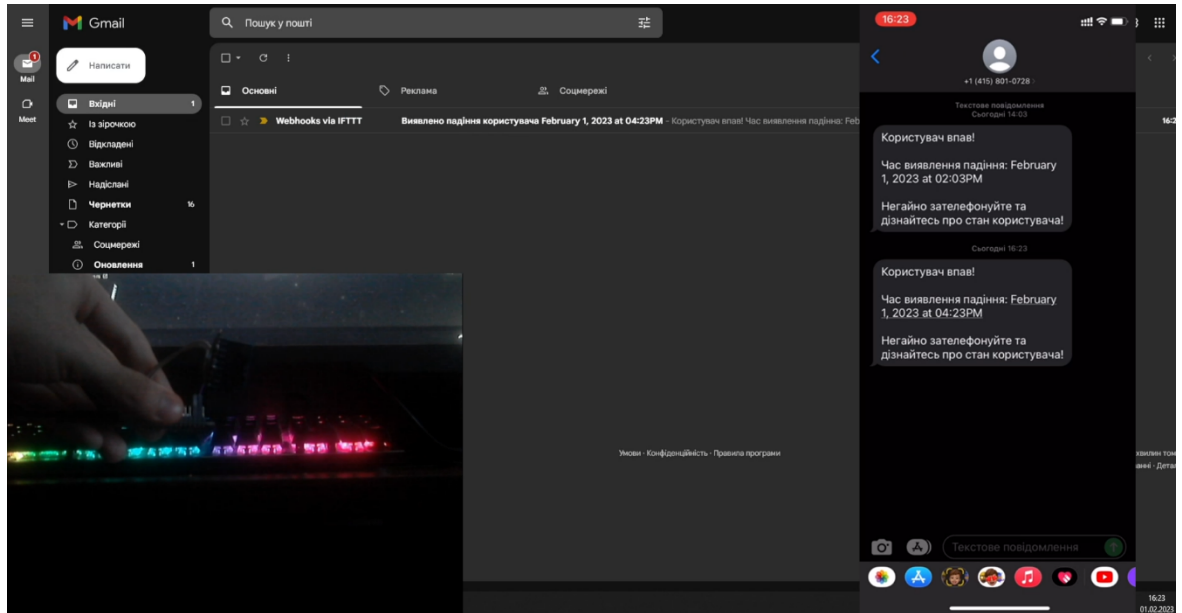


Рисунок 4.21 – Результат роботи

#### 4.4 Порівняння споживання до та після оптимізації

Під час виконання роботи, було зроблено дві версії програмної частини. З оптимізацією споживання апаратної платформи та без. Оптимізація була досягнута завдяки відключенню Wi-Fi модуля під час ввімкнення платформи, оскільки він використовується тільки для відправки даних до сервісу IFTTT. Тож було вирішено вмикати модуль Wi-Fi після виявлення падіння. На рис. 4.22 та рис. 4.23 зображено графіки споживання до та після оптимізації відповідно.

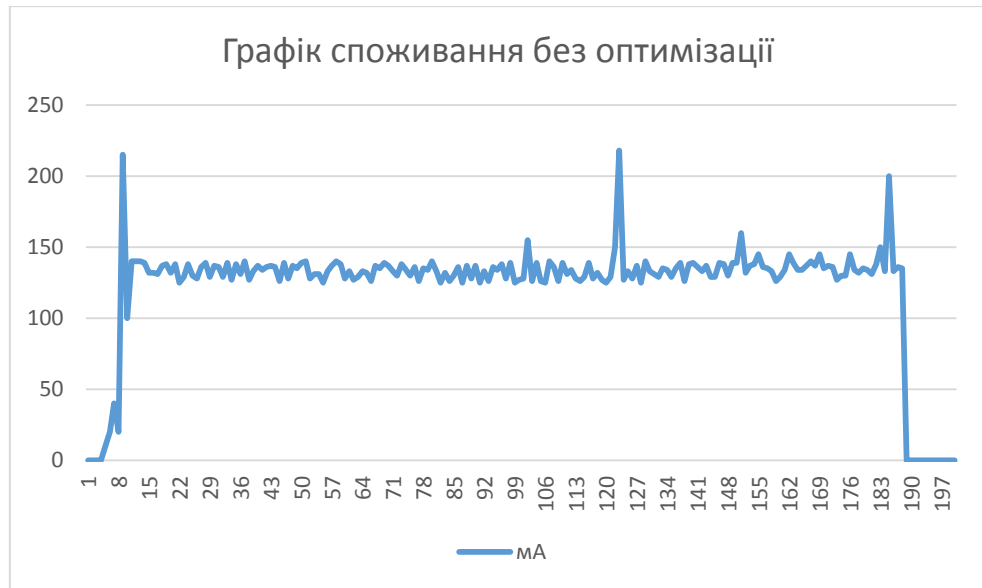


Рисунок 4.22 – Графік споживання без оптимізації



Рисунок 4.23 – Графік споживання після оптимізації

На рис. 4.23 можна побачити, що споживання знижується до 80 мА без ввімкненого модуля Wi-Fi, після того як виявляється падіння, вмикається модуль, з'єднується з мережею та відправляє запит до IFTTT. Після завершення, модуль вимикається і пристрій продовжує роботу.

#### 4.5 Порівняння прототипу з існуючим рішенням

Наприкінці було порівняно створений прототип з вже існуючим рішенням у вигляді смарт-годинника Apple Watch SE2, оскільки його характеристики можна знайти у відкритому доступі.

У табл. 4.1 наведено порівняльну таблицю характеристик прототипу та смарт-годинника Apple Watch SE2. Були обрані певні характеристики які можна віднести до обох пристроїв.

Таблиця 4.1 – Порівняння характеристик пристроїв

Характеристика	Прототип програмно-апаратного комплексу	Apple Watch SE2
Бездротовий зв'язок	802.11 b/g/n	LTE, 802.11 b/g/n, Bluetooth 5.3
Живлення	2400 мА/год	245 мА/год
Особливості	Виявлення падіння за допомогою модуля акселерометра та гіроскопа	Наявно: компас, висотомір, оптичний датчик серця, екстрений виклик, високоточний акселерометр для виявлення падінь та аварій, гіроскоп з динамічним діапазоном, датчик зовнішньої освітленості, динамік, мікрофон
Ціна	20\$	249\$

Apple Watch SE2 є більш універсальним пристроєм для повсякденного життя, тому в ньому наявна більша кількість сенсорів та технологій, ніж у спроектованому прототипі який сфокусований на певній задачі, а саме виявленні падіння.

## Висновки до розділу 4

На початку четвертого розділу було виконано налаштування маршрутизатора для успішного з'єднання платформи з мережею Wi-Fi, та налаштовано сервіс інтернет речей IFTTT (If This Then That). З персонального комп'ютера було створено аплет для надсилання Email повідомлення користувачу, а з мобільного застосунку було створено аплет для надсилання SMS повідомлення на зареєстрований номер телефону.

Далі були проведені експерименти для визначення верхньої та нижньої межі величини прискорення та кутової швидкості. В результаті експериментів, було отриману нижню межу величини прискорення у 2,5 g, верхню межу у 3,5 g. Діапазон кутового прискорення становив від 350 до 450 град/с. Під час перевірки працездатності системи, після імітації падіння, через декілька секунд апаратна платформа під'єдналась до Wi-Fi мережі, надіслала 2 запити до IFTTT, та сервіс IFTTT надіслав повідомлення на вказаний Email та мобільний номер. Також було наведено 2 графіки споживання апаратної платформи до та після оптимізації. Оптимізацію було досягнуто завдяки вимкненню Wi-Fi модуля до моменту відправки повідомлення про падіння.

Наприкінці була наведена порівняльна таблиця спроектовано прототипу та смарт-годиннику Apple Watch SE2. Apple Watch SE2 є більш універсальним пристроєм для повсякденного життя, тому в ньому наявна більша кількість сенсорів та технологій, ніж у спроектованому прототипі який сфокусований на певній задачі, а саме виявленні падіння.



## ВИСНОВКИ

Під час виконання кваліфікаційної роботи магістра було проаналізовано матеріал щодо можливих технологій та методів виявлення падіння, можливі апаратні платформи для реалізації прототипу. Спроектовано апаратний комплекс на базі LoLin NodeMCU v3 ESP8266 та 6-осьового модуля акселерометра та гіроскопа GY-521 для виявлення падіння в реальному часі та розроблено програмну частину на мові програмування C для отримання даних з датчиків, їх обробки та надсилання попередження до користувача за допомогою сервісу IFTTT.

Результатом виконання став пристрій, який здатний протягом 30 годин завдяки методам оптимізації споживання, вимірювати значення величини прискорення та кутової швидкості і використовувати ці дані для визначення чи трапилось падіння. Якщо програма фіксує падіння, вона надсилає запит на сервер IFTTT, який в свою чергу надсилає повідомлення користувачу по Email та SMS.

В той самий час, основним недоліком є недостатня ємність батареї. LiSOC12 батареї ємності 2400 мА/год вистачає лише на 30 годин безперервної роботи. В подальшому планується замінити на більш дорогий та ефективний метод живлення комплексу, додати LTE модуль, більш точніший модуль акселерометра та гіроскопа та спроектувати і зробити корпус для пристрою.

Результати виконання магістерської роботи можуть бути рекомендовані для використання у повсякденному житті, наприклад для людей похилого віку, щоб надавати інформації про можливе падіння їх опікуну або піклувальнику.

В процесі виконання також було виконано частину з охорони праці. Були проаналізовані умови в приміщенні підприємства в якому проводились дослідження. Було запропоновано доцільні заходи щодо його покращення:

облаштування поверхонь інтер'єру з максимально можливим коефіцієнтом відбиття.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. WHO global report on falls prevention in older age. URL : <https://bit.ly/3UBt9AY> (дата звернення : 07.12.2022).
2. What is the internet of things (IoT)?. URL: <https://bit.ly/3tAjYpr> (дата звернення: 07.12.2022).
3. Волощук С. І., Савінов В. Ю. Застосування UWB-модуля DW1000 у робототехнічних системах. Могілянські читання – 2022 : тези доп. XXV Всеукр. наук.-метод. конф. Миколаїв, 07–11 листоп. 2022 р. Миколаїв : Чорном. нац. ун-т ім. Петра Могили, 2022. С. 63–65.
4. Parikh R, Mathai A, Parikh S, Sekhar GC, Thomas R. Understanding and using sensitivity, specificity and predictive values. *Indian J Ophthalmol.* 2008. Вип. 56, № 1, С. 45–50.
5. Rougier C, Meunier J, St-Arnaud A, Rousseau J. Robust video surveillance for fall detection based on human shape deformation. *IEEE Transactions on Circuits and Systems for Video Technology.* 2011. Вип. 21, № 5, С. 611-622.
6. Miaou SG, Sung PH, Huang CY. A customized human fall detection system using omni-camera images and personal information. *Proceedings of the 1st Distributed Diagnosis and Home Healthcare Conference.* 2006. С. 39-42.
7. Fu Z, Delbruck T, Lichtsteiner P, Culurciello E. An address-event fall detector for assisted living applications. *IEEE Transactions on Biomedical Circuits and Systems.* Вип. 2, № 2, С. 88-96.
8. Hazelhoff L, Han J, de With PHN. Video-based fall detection in the home using principal component analysis. *Proceedings of the 10th International Conference on Advanced Concepts for Intelligent Vision Systems.* 2008. С. 298-309.
9. Mastorakis G, Makris D. Fall detection system using Kinect's infrared sensor. *Journal of Real-Time Image Processing.* 2012. С. 1-12

10. Cucchiara R, Rita H, Prati A, Andrea O, Vezzani R, Roberto C. A multi-camera vision system for fall detection and alarm generation. *Expert Systems*. 2007. Вип. 24, № 5, С. 334-345.
11. Що таке модель суміші гауса (гмм)? URL : <https://bit.ly/3uBSvUC> (дата звернення : 11.12.2022).
12. Приховані марковські моделі. URL : <https://bit.ly/3he545B> (дата звернення : 11.12.2022).
13. Fuzzy Logic: Definition, Meaning, Examples, and History. URL : <https://bit.ly/3VFXijO> (дата звернення : 11.12.2022).
14. Kerdegari H, Samsudin K, Ramli AR, Mokaram S. Evaluation of fall detection classification approaches. *Proceedings of the 4th International Conference on Intelligent and Advanced Systems*. 2012. С. 131-136.
15. Plaza I, Martin L, Martin S, Medrano C. Mobile applications in an aging society: Status and trends. *Journal of Systems and Software*. 2011. Вип. 84, № 11, С. 1977-1988.
16. Santos G. L., Endo P. T., Monteiro K. H. de C., Rocha E. da S., Silva I., Lynn T. Accelerometer-based human fall detection using convolutional neural networks. *Sensors (Switzerland)*. 2019. Вип. 19, № 7.
17. Palmerini L., Klenk J., Becker C., Chiari L. Accelerometer-based fall detection using machine learning: Training and testing on real-world falls. *Sensors (Switzerland)*. 2020. Вип. 20, № 22.
18. Khojasteh S. B., Villar J. R., Chira C., González V. M., la Cal E. de. Improving fall detection using an on-wrist wearable accelerometer. *Sensors (Switzerland)*. 2018. Вип. 18, № 5.
19. Video-based fall risk assessment system: пат. US20200205697A1 United States. № US16/731,025; заявл. 02.07.2020.
20. Detecting falls using a mobile device: пат. US11282363B2 United States. № US16/929,028; заявл. 29.10.2020.

21. Fall detection systems and methods: пат. US10485452B2 United States. № US16/034,266; заявл. 22.11.2018.

22. Introduction to MEMS Accelerometers. PCB Piezotronics | Sensors to measure vibration, acoustics, force, pressure, load, strain, shock & torque. URL: <https://www.pcb.com/resources/technical-information/mems-accelerometers> (дата звернення: 24.12.2022).

23. MEMS gyroscope provides precision inertial sensing in harsh, high temperature environments | analog devices. Mixed-signal and digital signal processing ICs | Analog Devices. URL: <https://www.analog.com/en/technical-articles/mems-gyroscope-provides-precision-inertial-sensing.html> (дата звернення: 24.12.2022).

24. Mechanics - Coriolis force. Encyclopedia Britannica. URL: <https://www.britannica.com/science/mechanics/Coriolis-force> (дата звернення: 24.12.2022).

25. Літій-Тіоне-Хлоридні Промисл батареї 3,6 V Li-SOCL<sub>2</sub> Inorganic( Спец. літій). Товари та послуги компанії "Інтернет-магазин "Батарейка"". "Інтернет-магазин "Батарейка"". URL: <https://betery.com.ua/ua/g3426392-litij-tionil-hloridnye> (дата звернення: 25.12.2022).

26. Welcome to fritzing. Welcome to Fritzing. URL: <https://fritzing.org/> (дата звернення: 25.12.2022).

27. What is arduino software (IDE), and how use it ? - AndProf. AndProf. URL: <https://andprof.com/tools/what-is-arduino-software-ide-and-how-use-it/> (дата звернення: 10.01.2023).

28. Wire – Arduino Reference. Arduino – Home. URL: <https://www.arduino.cc/reference/en/language/functions/communication/wire/> (дата звернення: 10.01.2023).

29. GitHub - ekstrand/ESP8266wifi: ESP8266 Arduino library with built in reconnect functionality. GitHub. URL: <https://github.com/ekstrand/ESP8266wifi> (дата звернення: 10.01.2023).

30. What is IFTTT? How to use If This, Then That services. Computerworld. URL: <https://www.computerworld.com/article/3239304/what-is-ifttt-how-to-use-if-this-then-that-services.html> (дата звернення: 10.01.2023).

## ДОДАТОК А

### Програмний код апаратної частини для виявлення падіння

```
#include <Wire.h> // Бібліотека Wire
#include <math.h> // Бібліотека math
#include <ESP8266WiFi.h> // Бібліотека ESP8266WiFi
#define THRESHOLD_LOW 2.0 // Нижня межа
#define THRESHOLD_HIGH 3.5 // Верхня межа
const int addr=0x68; // I2C адреса GY-51
int16_t AccelX,AccelY,AccelZ,TempI,gyroX,gyroY,gyroZ; // Змінні для зберігання
даних з GY-521
int accel_offset_x, accel_offset_y, accel_offset_z; // Дані калібровки
акселерометра
int gyro_offset_x, gyro_offset_y, gyro_offset_z; // Дані калібровки гіроскопа
float accx=0, accy=0, accz=0, gyrx=0, gyry=0, gyrz=0; // Дані отримані після
калібровки
int gyro_angle = 0; // Кутова швидкість

boolean fallDetected = false; // булеве значення якщо було виявлено падіння
boolean tr_lower=false; // булеве значення якщо була перетнута нижня межа
boolean tr_higher=false; // булеве значення якщо була перетнута верхня межа
boolean tr_gyroAngle=false; // булеве значення якщо було перетнуто межу
кутового прискорення
byte tr_lower_count=0; // лічильник для нижньої межі
byte tr_higher_count=0; // лічильник для верхньої межі
byte tr_gyroAngle_count=0; // лічильник для межі кутової швидкості
const char *ssid = "voloshchuk-serhii"; // Ім'я Wi-Fi мережі
const char *pass = "voloshchuk-serhii"; // Пароль від мережі Wi-Fi
const char *host = "maker.ifttt.com"; // Сервер IFTTT
const char *userInfo = "voloshchuk-serhii-diploma"; // Таємний ключ від IFTTT
void setup()
{
  WiFi.mode( WIFI_OFF ); // Вимикаємо Wi-Fi модуль для зменшення споживання
  WiFi.forceSleepBegin();
  delay( 1 );
  WiFi.persistent( false ); // Вимикаємо завантаження налаштувань з FLASH
пам'яті
  Serial.begin(115200);
  Wire.begin();
  Wire.beginTransmission(addr);
```

```

Wire.write(0x6B); // Доступ до реєстру GY-521 для його вмикання
Wire.write(0);
Wire.endTransmission(true);
Serial.println("Розпочато процес калібрації модуля GY-521");
calibrate_mpu();
Serial.println("Процес калібрації завершено!");
}
void loop() {
  mpu_read();
  // Перевод даних з акселерометра у одиниці виміру g
  accx = (AccelX-accel_offset_x)/16384.00;
  accy = (AccelY-accel_offset_y)/16384.00;
  accz = (AccelZ-accel_offset_z)/16384.00;
  // Перевод даних з акселерометра у одиниці виміру град/с
  gyrx = (gyroX-gyro_offset_x) / 131.07;
  gyry = (gyroY-gyro_offset_y) / 131.07;
  gyrz = (gyroZ-gyro_offset_z) / 131.07;
  // Обчислення величини прискорення акселерометра
  float acc_magnitude = sqrt(accx * accx + accy * accy + accz * accz);
  // Обчислення величини кутової швидкості
  gyro_angle = sqrt(gyrx * gyrx + gyry * gyry + gyrz * gyrz);
  if (acc_magnitude>=THRESHOLD_LOW && tr_higher==false){ // Умова перетинання
нижньої межі
    tr_lower=true;
    Serial.println("Нижня межа була перетнута");
  }
  if (tr_lower==true){
    tr_lower_count++;
    if (acc_magnitude>=THRESHOLD_HIGH){ // Умова перетинання верхньої межі
      tr_higher=true;
      Serial.println("Верхня межа була перетнута");
      tr_lower=false; tr_lower_count=0;
    }
  }
  if (tr_higher==true){
    tr_higher_count++;
    gyro_angle = sqrt(gyrx * gyrx + gyry * gyry + gyrz * gyrz);
    if (gyro_angle>=350 && gyro_angle<=440){ // Умова при якій кутова швидкість
була у заданих межах
      tr_gyroAngle=true; tr_higher=false; tr_higher_count=0;
      Serial.println("Кутова швидкість була заданих межах");
    }
  }
}

```



```

    }
}
if (tr_gyroAngle==true){
    tr_gyroAngle_count++;
    if (tr_gyroAngle_count>=10){
        delay(1000);
        gyro_angle = sqrt(gyrx * gyrx + gyry * gyry + gyrz * gyrz);

        if ((gyro_angle>=0) && (gyro_angle<=10)){ // Якщо кутова швидкість
гіроскопа залишається незмінною, спрацьовує повідомлення користувача
            fallDetected=true; tr_gyroAngle=false; tr_gyroAngle_count=0;
        }
        else{ // В іншому випадку, програма починає знову обчислювати дані з
модуля

            tr_gyroAngle=false; tr_gyroAngle_count=0;
            Serial.println("Користувач не впав, повернення до обчислень");
        }
    }
}
if (fallDetected==true){
    WiFi.forceSleepWake(); // Вмикаємо Wi-Fi
    delay( 1 );

    WiFi.mode( WIFI_STA );
    WiFi.begin( ssid, password );
    while (WiFi.status() != WL_CONNECTED) // З'єднуємось поки з'єднання не буде
успішним
    {
        delay(500);
        Serial.print(".");
    }

    Serial.println("Користувач впав!");
    // Надсилаємо повідомлення по пошті на СМС користувачу
    send_event("fall_detected_email");
    send_event("fall_detected_sms");
    fallDetected=false;
}
if (tr_higher_count>=6){ // Час для можливості значно змінити величини
кутової швидкості
    tr_higher=false; tr_higher_count=0;
}

```

```
    }
    if (tr_lower_count>=6){ // Час для можливості перетнути верхню межу
        tr_lower=false; tr_lower_count=0;
    }
    delay(100);
    WiFi.mode( WIFI_OFF ); // Вимикаємо Wi-Fi модуль для зменшення споживання
    WiFi.forceSleepBegin();
    delay(1);
}
void mpu_read(){
    Wire.beginTransmission(addr);
    Wire.write(0x3B);
    Wire.endTransmission(false);
    Wire.requestFrom(addr,14,true); // Звернення до 14 реєстрів модуля
    // Зчитування даних з реєстрів GY-521
    AccelX=Wire.read()<<8|Wire.read();
    AccelY=Wire.read()<<8|Wire.read();
    AccelZ=Wire.read()<<8|Wire.read();
    TempI=Wire.read()<<8|Wire.read();
    gyroX=Wire.read()<<8|Wire.read();
    gyroY=Wire.read()<<8|Wire.read();
    gyroZ=Wire.read()<<8|Wire.read();
}
void calibrate_mpu() {
    for(int i=0;i<1000;i++){
        Wire.beginTransmission(addr);
        Wire.write(0x3B);
        Wire.endTransmission(false);
        Wire.requestFrom(addr,14,true);
        AccelX=Wire.read()<<8|Wire.read();
        AccelY=Wire.read()<<8|Wire.read();
        AccelZ=Wire.read()<<8|Wire.read();
        TempI=Wire.read()<<8|Wire.read();
        gyroX=Wire.read()<<8|Wire.read();
        gyroY=Wire.read()<<8|Wire.read();
        gyroZ=Wire.read()<<8|Wire.read();

        accel_offset_x += AccelX;
        accel_offset_y += AccelY;
        accel_offset_z += AccelZ;
        gyro_offset_x += gyroX;
```

```
    gyro_offset_y += gyroY;
    gyro_offset_z += gyroZ;
    delay(2);
}
// Обчислення фінального значення зміщення калібрації
accel_offset_x /= 1000;
accel_offset_y /= 1000;
accel_offset_z /= 1000;
gyro_offset_x /= 1000;
gyro_offset_y /= 1000;
gyro_offset_z /= 1000;
}
void send_event(const char *event)
{
    WiFiClient client;
    const int httpPort = 80;
    if (!client.connect(host, httpPort)) {
        Serial.println("Не вдалось з'єднатись з сервером IFTTT");
        return;
    }

    String url = "/trigger/" + event + "/with/key/" + userInfo // Створюємо
унікальне посилання для GET запросу
    // Надсилаємо GET запит на сервер
    client.print(String("GET ") + url + " HTTP/1.1\r\n" + "Host: " + host +
"\r\n" + "Connection: close\r\n\r\n");
    while(client.connected())
    {
        if(client.available())
        {
            String line = client.readStringUntil('\r');
            Serial.print(line);
        } else {
            // Очікуємо на дані
            delay(50);
        }
    };
}
client.stop();
}
```