

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

СІРЯК ЮЛІЯ ВОЛОДИМИРІВНА

Допускається до захисту:  
Завідувач кафедри  
інформаційних технологій,  
к.т.н., доцент  
\_\_\_\_\_ Т. В. Нескородева  
«\_\_» \_\_\_\_\_ 20\_\_р.

БЕЗПРОВІДНА ПЕРЕДАЧА ДАНИХ У ТРЕНАЖЕРІ  
«БОКСЕРСЬКА ГРУША»

Спеціальність 105 Прикладна фізика та наноматеріали

Кваліфікаційна (бакалаврська) робота

Керівник:  
Чернов Д. В., доцент кафедри  
інформаційних технологій  
к.т.н.

\_\_\_\_\_  
(підпис)

Оцінка : \_\_\_\_\_ / \_\_\_\_\_ / \_\_\_\_\_  
(бали за шкалою ЄКТС/за національною шкалою)

Голова ЕК: \_\_\_\_\_  
(підпис)

Вінниця 2021

## АНОТАЦІЯ

**Сіряк Ю. В. Безпроводна передача даних у тренажері «боксерська груша».** Спеціальність 105 «Прикладна фізика та наноматеріали», Освітня програма «Технології інтернету речей», Донецький національний університет імені Василя Стуса, Вінниця, 2021.

У кваліфікаційній (бакалаврській) роботі запропоновано та побудовано систему збору даних у спортивному тренажері для підвищення мобільності, об'єктивності та точності вимірювання параметрів удару боксера. На основі проведеного аналізу було обрано найбільш оптимальне рішення для збору та передачі даних з тренажера.

Ключові слова: Wi-Fi, боксерська груша, канал зв'язку, тренажер, мережа, ESP32, MPU6050.

Табл. 2. Рис. 23. Бібліограф.: 21 найм.

## ABSTRACT

**Siriak Y. V. Wireless data transmission in punching bag trainer.** Speciality 105 «Applied Physics and Nanomaterials», Programme «Internet of Things Technology». Vasyl' Stus Donetsk National University, Vinnytsia, 2021.

In the qualification (bachelor's) work the system of data collection in the sports simulator for increase of mobility, objectivity and accuracy of measurement of parameters of blow of the boxer is offered and constructed. Based on the analysis, the most optimal solution for collecting and transmitting data from the simulator was selected.

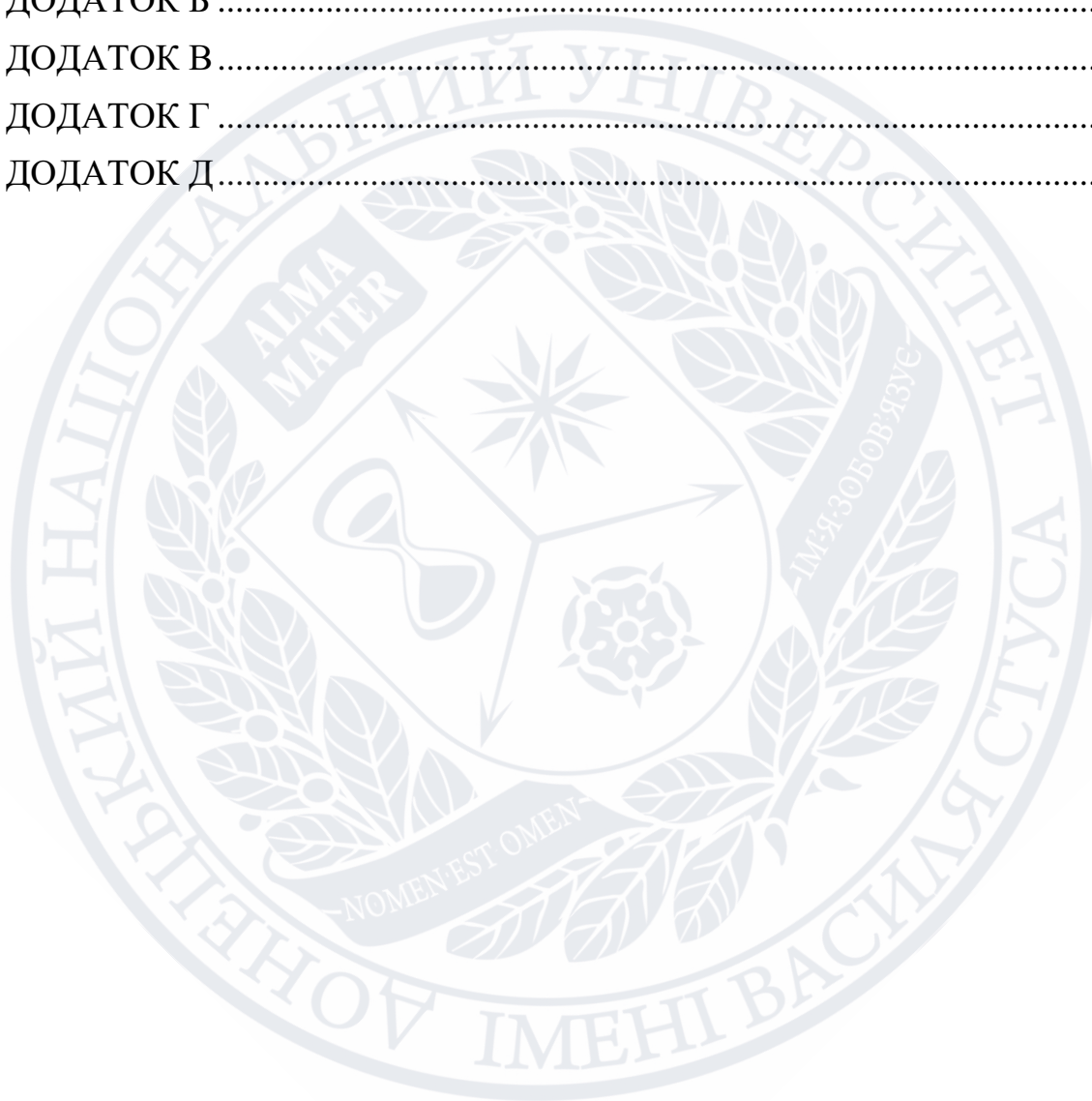
Key words: Wi-Fi, punching bag, communication channel, trainer, network, ESP32, MPU6050.

Tabl. 2. Fig. 23. Bibliography: 21 items

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	5
ВСТУП .....	7
1.1 Порівняння мережевих технологій для бездротового зв'язку .....	10
1.1.1 LoRaWAN .....	10
1.1.2 Стриж .....	11
1.1.3 SigFox .....	11
1.1.4 Wheitghless.....	12
1.1.5 Nuel.....	12
1.1.6 Nwave .....	13
1.1.7 Dash7 .....	14
1.1.8 GSM, LTE.....	14
1.1.9 NB-LTE і NB-CIoT.....	15
1.1.10 Wireless RF.....	16
1.1.11 Bluetooth Low Energy (BLE).....	16
1.1.12 Wi-Fi .....	16
1.1.13 Wi-Fi HaLow .....	18
1.1.14 Thread .....	18
1.1.15 ZigBee.....	19
1.1.16 Z-Wave.....	20
1.2 Топології бездротових мереж .....	21
1.2.1 Топологія точка-точка .....	21
1.2.2 Топологія «Зірка».....	21
1.2.3 Топологія «Багато осередкова мережа» .....	22
1.2.4 Топологія «кластерне дерево».....	22
1.3 Порівняння мережевих топологій .....	22
1.4 Типова структура мережі Wi-Fi.....	25
РОЗДІЛ 2. МОДУЛЬ ПЕРЕДАЧІ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ Wi-Fi	29
2.1 Побудова каналу зв'язку за допомогою мікроконтролера ESP32 .....	29
2.2 Трьохосьовий гіроскоп з трьохосьовим акселерометром MPU6050 .....	35
2.3 Модуль GY-521, трьохосьовий гіроскоп з трьохосьовим акселерометром MPU6050 на базі мікроконтролера ESP32.....	38

3.1 Опис боксерської груші.....	40
3.2 Огляд проекту.....	42
3.2 Як працює тренажер для боксу.....	52
ВИСНОВКИ.....	54
СПИСОК ВИКОРИСТАНИХ ПОСИЛАНЬ .....	55
ДОДАТОК А.....	57
ДОДАТОК Б .....	62
ДОДАТОК В.....	64
ДОДАТОК Г .....	66
ДОДАТОК Д.....	69





## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

**ZigBee** – ZigBee (Специфікація мережевих протоколів верхнього рівня)

**Wi-Fi** – Wireless Fidelity (Бездротова локальна мережа)

**Bluetooth** – Bluetooth (Блютус)

**NFC** – Near Field Communication (Комунікація ближнього поля)

**LoRaWAN** – Long Range Wide Area Network (Глобальна мережа далекого радіусу дії)

**SigFox** – (Глобальна мережа для Інтернету речей (IoT))

**UNB** – Ultra Narrow Band (Протокол бездротової передачі даних для високоемкий мереж на основі сверхузкополосной зв'язку)

**LPWAN** – Low – Power Wide – Area Network (Мережа енергоефективна, далекого радіусу дії)

**ISM** – Information Security Management (Управління інформаційною безпекою)

**IoT** – Internet of Things (Інтернет речей)

**BLE** – Bluetooth Low Energy (Бездротова технологія Bluetooth з низьким енергоспоживанням)

**BSIG** – Bluetooth Special Interest Group (Організація по стандартизації, яка спостерігає за розробкою стандартів Bluetooth і ліцензуванням технологій і товарних знаків Bluetooth для виробників)

**WPA2** – Wireless Protected Access ver. 2.0 (Оновлена програма сертифікації бездротового зв'язку)

**ARM** – Advanced RISC Machine (Мікропроцесорна архітектура RISC–(RISC), що розробляється ARM Limited)

**TNP** – Thread Networking Protocol (Бездротової мережевий протокол на основі IP)

**6LoWPAN** – IPV6 over Low power Wireless Personal Area Networks (Персональна мережа малої потужності)

**IP** – Internet Protocol (Міжмережевий протокол)

**IPv6** – Internet Protocol version 6 (Нової версії 6 інтернет протокол)

**ANT** – Another Neat Too (Утиліта для автоматизації процесу складання програмного продукту)

**RFID** – Radio Frequency IDentification (Радіочастотна ідентифікація)

**UWB** – **Ultra Wideband** (Бездротова технологія зв'язку на малих відстанях при низьких витратах енергії)

**IDEN** – Integrated Digital Enhanced Network (Технологія мобільного зв'язку фірми Motorola)

**WiMAX** – Worldwide Interoperability for Microwave Access (Технологія бездротового широкосмугового доступу в інтернет)

**GSM** – Global System for Mobile Communications (Глобальна система для рухомого зв'язку)

**GPRS** – General Packet Radio Service (Пакетна радіозв'язка загального користування)

**SPI** – Serial Peripheral Interface (Послідовний периферійний інтерфейс)

**ZDO** – ZigBee Device Objects (Об'єкти пристрою ZigBee)

**AF** – Application Framework (Фреймворк додатків)

**CSMA** – Carrier Sense, Multiple Access (Імовірнісний мережевий протокол канального рівня)

**ШИМ** – Широтно-імпульсна модуляція або модуляція за тривалістю імпульсів

## ВСТУП

На обладнанні різних поколінь і технологій засновані сучасні мережі зв'язку [1]. Синонімом для лінії зв'язку вживають термін канал зв'язку. По лініях зв'язку передаються електричні сигнали, апаратури передачі даних, а також проміжної літератури [2]. Основні вимоги до каналів зв'язку — висока швидкість, надійність та якість [3]. Існують провідні, кабельні, безпроводні лінії зв'язку. Бездротові канали зв'язку забезпечують швидке і комфортне підключення до мережі, які зараз дуже широко розвиваються і модернізуються [4].

Уже кілька десятиліть люди застосовують комп'ютерні мережі для забезпечення зв'язку між персоналом, комп'ютерами й серверами в компаніях, коледжах і містах. Однак спостерігається тенденція де все більш широко використовуються бездротові мережі. І дійсно, зараз доступні бездротові інтерфейси, що дозволяють використовувати мережеві служби, працювати з електронною поштою і переглядати Web-сторінки незалежно від того, де перебуває користувач. Ці бездротові додатки дозволяють людям "розширити" своє робоче місце та отримати в результаті цього ряд переваг [4].

Бездротові мережі відіграють важливу роль в житті людей, де б вони не знаходилися — на роботі, вдома або в громадському місці. Відрізняються вони невеликими відстанями передачі (до 17 м, або 50 футів), для розгортання в невеликому приміщенні або в "персональній зоні" робить їх ідеальними [4].

Існує безліч різновидів бездротового зв'язку, але найважливішою особливістю бездротових мереж є те, що зв'язок здійснюється між комп'ютерними пристроями. До них відносяться персональні цифрові помічники (personal digital assistance, PDA), ноутбуки, персональні комп'ютери (ПК), сервери та принтери. Комп'ютерними пристроями вважаються такі, які мають процесори, пам'ять і засоби взаємодії з якоюсь мережею. Бездротові мережі дозволяють людям зв'язуватися та отримувати доступ до прикладних програм та інформації без використання дротяних з'єднань. Це забезпечує свободу



пересування та можливість використання додатків, що знаходяться в інших частинах будинку, міста або у віддаленому куточку світу [4].

Тому для боксерської груші доцільно зробити безпроводний канал зв'язку. Боксерська груша — це тренажер для боксу та інших видів єдиноборств належати до спорту і стосується конструкції навчально–тренувальних снарядів для контролю і відпрацювання ударів в боксі та інших видах силових єдиноборств [4].

**Актуальність теми.** Моніторинг фізичних параметрів спортсменів є однією з важливих задач при тренуванні. Це дозволяє проводити відбір найбільш здатних спортсменів до відповідного виду спорту. Для здійснення контролю таких фізичних параметрів, як, наприклад, сила удару боксера по груші, швидкість, прискорення та інші, потрібен канал зв'язку, який дозволить передавати інформацію з датчиків до терміналу, де ці параметри зображуються. Тому, задача побудови каналу зв'язку для моніторингу фізичних параметрів спортсмена при тренуванні є актуальною. Так, як спортсмен постійно рухається, цей канал доцільно зробити бездротовим.

**Метою дослідження** є побудова системи збору даних у спортивному тренажері для підвищення мобільності, об'єктивності та точності вимірювання параметрів удару боксера.

**Завдання дослідження:**

- проаналізувати сучасні підходи бездротової передачі даних по вузькосмуговим каналам зв'язку;
- обрати за проведеним аналізом найбільш оптимальне рішення для збору та передачі даних з тренажера;
- побудова каналу бездротового зв'язку для передачі даних з датчиків за обраною технологією у тренажері «Боксерська груша».

**Об'єктом дослідження** є процес передачі інформації у безпроводному каналі зв'язку.

**Предметом дослідження** є засоби передачі даних у безпроводному каналі зв'язку інтернету речей.



## РОЗДІЛ 1. ПОРІВНЯННЯ МЕРЕЖЕВИХ ТЕХНОЛОГІЙ ДЛЯ БЕЗДРОВОГО ЗВ'ЯЗКУ

Різноманітні бездротові пристрої висувають різні вимоги до вживаного радіоканалу. В одних випадках потрібна максимальна швидкість передачі даних, в інших ключовим параметром стають рівень споживання або дальність зв'язку. З цієї причини можна спостерігати співіснування цілого ряду різних технологій, наприклад, ZigBee, Wi-Fi, Bluetooth і інших. Компанія Texas Instruments пропонує широкий вибір рішень для найбільш популярних бездротових технологій, в тому числі – унікальні мікроконтролери сімейств CC3xx, CC2xx, CC1xx, CC430 і RF430 з вбудованими модулями радіопередавачів. З їх допомогою вдається максимально спростити процес розробки час виведення на ринок нових пристроїв знано скоротити [5].

В сучасних бездротових додатках до каналу передачі даних пред'являються такі вимоги як:

- підтримка однієї з популярних топологій мережі («точка–точка», «зірка», Mesh–мережі);
- висока пропускна здатність до десятків Мбіт/с;
- максимальний радіус дії від кількох сантиметрів (NFC) до декількох кілометрів (пропрієтарні протоколи Sub1GHz);
- мале споживання (від десятків міліампер в активному режимі до десятих часток мікроампер в режимі сну);
- високий рівень безпеки (підтримка шифрування, функцій хешування, протоколів аутентифікації й так далі);
- конкурентна ціна (мінімальна кількість і вартість компонентів);
- малі габарити (особливо важливо для портативних додатків);
- простота реалізації апаратної та програмної частин [5].

Більшість вимог з вищенаведеного списку суперечить одне одному. З цієї причини зараз застосовують різні бездротові технології, кожна з яких є оптимальною для конкретного додатка або пристрою [5].

## 1.1 Порівняння мережевих технологій для бездротового зв'язку

Існує дуже велика кількість бездротових технологій, кожна з яких має свої особливості. Бездротові технології поділяються на далекий радіус дії, які наведені у таблиці А.1.1 та близький радіус дії в таблиці А.1.2 (Додаток А) [5].

### 1.1.1 LoRaWAN

LoRaWAN або Long Range Wide Area Network представлена дослідним центром IBM Research і компанією Semtech як енергоефективна мережева технологія. Технологія базується на Semtech LoRa (™) РНУ чипі. Архітектура мережі виглядає як зірка, кінцеві пристрої підключаються по бездротовому зв'язку до одного або кількох шлюзів, а шлюзи підключаються до мережевого сервера через стандартну IP-з'єднання [5].

Переваги LoRa:

- відкритий стандарт та велика дальність;
- висока проникаюча здатність в міській забудові;
- низьке енергоспоживання, за оцінками до 10 років роботи сенсора від батареї AA;
- різні неліцензовані частоти, такі як 915 МГц суб-ГГц ISM смуг, 868 МГц, 433 МГц, 109 МГц;
- адаптивна швидкість передачі даних;
- підтримує особисті та громадські мережі;
- комплексна безпека і вбудовані ідентифікація і аутентифікація.

Недоліки LoRa:

- низька швидкість передачі даних;
- Semtech – єдиний постачальник чіпів;

- відсутній роумінг [5].

Концентратори для LoRa поставляються такими компаніями, як MultiTech, і вже створені громадські мережі, як наприклад The Things Network [5].

### 1.1.2 Стриж

Система реалізовується російською компанією Стриж – Телематика, використовується власний протокол Marcato 2.0. Частота може бути адаптована під ISM діапазон. Технологія до певної міри схожа з технологією LoRa з усіма плюсами і мінусами останньої. Принципова відмінність: у LoRa використовується широкосмугове кодування, а у Стриж – вузькополосна модуляція. За даними компанії, така модуляція дозволяє набагато ефективніше використовувати смугу спектра, збільшити чутливість і енергоефективність і знизити вартість [5].

### 1.1.3 SigFox

Система, заснована у Франції в 2009 році та побудована однойменною компанією. Ця технологія спочатку призначена для зв'язку на низьких швидкостях передачі даних [5].

Компанія проводить сертифікацію пристроїв SigFox Ready™.

Переваги SigFox:

- найбільше покриття та низька вартість;
- висока проникаюча здатність в міській забудові;
- наднизьким споживанням енергії;
- гнучкість в плані конструкції антени;
- протокол SigFox сумісний з існуючими трансиверами.

Недоліки SigFox:

- низька швидкість передачі даних;
- залежність від стільникового інфраструктури;
- обмежена стійкість [5].

#### 1.1.4 Weightless

Для обміну даними між базовою станцією і пристроями використовується технологія Weightless – група відкритих технологічних стандартів зв'язку LPWAN (Low-Power Wide-Area Network). Стандарти розвиваються некомерційною організацією Weightless SIG. Доступні 3 стандарти на даний час – Weightless-N, Weightless-P та Weightless-W [5].

Weightless-N використовує технологію Ultra Narrow Band (UNB), є стандартом однобічного зв'язку. З точки зору витрат, так і по енергоспоживанню в групі найекономічніший. Частоти TVWS (TV white space, невикористовувані частоти телевізійного спектра) використовує Weightless-W там, де це дозволено місцевими правилами. Це все здорожує термінал та збільшує його енергоспоживання [5].

Переваги Weightless:

- відкритий стандарт та велика дальність;
- висока проникаюча здатність в міській забудові;
- низьке енергоспоживання, за оцінками до 10 років роботи сенсора (Weightless-N);
- різні неліцензованому частоти (Weightless-P);
- підтримує особисті та громадські мережі;
- висока безпека та низька вартість (особливо Weightless-N).

Недоліки Weightless:

- низька швидкість передачі даних [5].

#### 1.1.5 Neul

Neul базується на протоколі Weightless, використовує неліцензовані ISM і TVWS частоти. У вересні 2014 року Neul була придбана компанією Huawei і стала дочірньою. Заявлено, що Neul і Huawei спільно працюють над новаторською технологією, яка дозволяє повторне використання мереж



мобільних операторів для широкого охоплення ультра–низької потужності зв'язку для додатків IoT [5].

Переваги Neul:

- велика дальність та висока проникаюча здатність в міській забудові;
- низьке енергоспоживання та добре поєднується з іншими стандартами на сусідніх частотах.

Недоліки Neul:

- низька швидкість передачі даних;
- пропрієтарна технологія [5].

#### 1.1.6 Nwave

Британська компанія з офісами в Лондоні, США і Данії очолюється випускником МФТІ Юрієм Бірченко. Технологія Nwave подібна до Neul, оскільки також заснована на протоколі Weightless, і її можна порівняти з SigFox, оскільки є пропрієтарною. Nwave іноді описують як VPN (virtual private network, віртуальна приватна мережа) всередині публічного трафіку з використанням стандарту Weightless–N. Використовується технологія Ultra Narrow Band (UNB) і неліцензованому ISM частоти. Компанія виробляє і реалізовує радіомодеми, базові станції, а також датчики з вбудованими модемами і набори для розробників [5].

Переваги Nwave:

- велика дальність та висока проникаюча здатність в міській забудові;
- дуже низьке енергоспоживання і підтримує особисті та громадські мережі;

- висока безпека та низька вартість.

Недоліки Nwave:

- низька швидкість передачі даних;
- пропрієтарна технологія [5].

### 1.1.7 Dash7

Dash7 Альянс протокол (або D7A) – це протокол бездротового зв'язку, який є відкритий та працює добре на частотах 433 МГц, 868 МГц і 915 МГц неліцензованому ISM / SRD діапазонів. Підтримується AES 128-бітове шифрування і до 167 кбіт / с передача даних, та максимальний пакет даних становить 256 байт. Протокол просувається некомерційним Альянсом Dash7 Alliance зі штаб-квартирою в Бельгії. Протокол заснований на міжнародному стандарті ISO / IEC 18000-7, що описує інтерфейс для активної RFID і використовуваним у військовій логістиці США (НАТО). Поточна версія протоколу DASH7 вже не відповідає стандартів ISO / IEC 18000-7 [5].

#### Переваги Dash7:

- стандарт – відкритий та підтримує велику дальність;
- проникаюча здатність висока в міській забудові;
- низьке енергоспоживання;
- різні неліцензованому частоти.

#### Недоліки Dash7:

- низька швидкість передачі даних;
- середня проникаюча здатність в воді;
- певні вимоги до антен [5].

### 1.1.8 GSM, LTE

Консорціум 3GPP (The 3rd Generation Partnership Project), який розробляє специфікації для мобільної телефонії, вже давно працює над поліпшенням GSM (Global System for Mobile Communications, спочатку Groupe Spécial Mobile) і LTE (Long-Term Evolution) з точки зору IoT. Це перш за все відповіді на виклики: проникаюча здатність, низьке енергоспоживання, економічність і масштабованість [5].

Переваги GSM, LTE:

- функціонування на існуючій інфраструктурі стільникових операторів;
- широке поширення в світі;
- висока швидкість передачі даних;
- підтримка особистих і суспільних мереж;
- висока комплексна безпека;
- роумінг.

Недоліки GSM, LTE:

- ліцензовані частоти;
- високі тарифи [5].

#### 1.1.9 NB–LTE і NB–CIoT

Корпорації Nokia Networks, Ericsson і Intel об'єдналися для просування технології Narrow–Band Long–Term Evolution (NB–LTE). Sprint, Verizon Wireless, Alcatel–Lucent, Qualcomm, Samsung, Sony і ZTE також стали частиною цієї ініціативи. NB–LTE розглядається деякими експертами як прямий виклик Huawei Technologies, яка розвиває технологію Narrowband Cellular IoT (NB–CIoT). NB–CIoT вже отримала підтримку таких важковаговиків, як Vodafone, T–Mobile, TeliaSonera і China Unicom [5].

Основна відмінність між NB–LTE і NB–CIoT зводиться до того, наскільки існуючі LTE–мережі можуть бути переорієнтовані на IoT. Huawei відмовився від коментарів на цей рахунок, але критики підходу «чистого аркуша» (clean slate) NB–CIoT відзначають, що ця технологія вимагає нових чіпсетів і, здається, не має зворотної сумісності з LTE–мережами старше Release 13. Переваги та недоліки цих технологій природним чином виростають з GSM і LTE [5].

### 1.1.10 Wireless RF

Бездротові радіо датчики (Wireless RF) та виконавчі механізми в розгортанні прості і дешевші. Ультранизьким енергоспоживанням характеризуються вони. Дальність дії досягає прямої видимості до 100 м та з зовнішніми антенами до 500м. Зазвичай працюють вони на частоті 315 або 433 МГц зі швидкістю 10 – 115.2 кбіт / с та підтримують AES шифрування 128 біт [5].

### 1.1.11 Bluetooth Low Energy (BLE)

Bluetooth Low Energy (BLE) є бездротової персональної мережевою технологією. За допомогою Bluetooth Special Interest Group, розроблена та реалізується. В даний момент технологія Bluetooth присутній на всіх мобільних платформах, BLE оснащуються мільйони нових пристроїв. Така технологія є надійною для ближніх комунікацій та дуже добре підтримується. Часто застосовується для зв'язку між смартфонами і іншими персональними, рідше домашніми електронними пристроями. На цій технології, зокрема, заснована технологія iBeacon [5].

Переваги BLE:

- широке поширення в світі;
- висока швидкість передачі даних;
- висока надійність.

Недоліки BLE:

- деякі проблеми з аутентифікацією і приватних даних;
- невисока проникаюча здатність в міській забудові;
- місцезнаходження пристрою не визначається [5].

### 1.1.12 Wi-Fi

Wi-Fi (або Wi-Fi, спочатку від англ. Wireless Fidelity) – це локальна бездротова мережева технологія, яка електронним пристроям дозволяє підключатися до мережі, з використанням частот 5 ГГц та 2,4 ГГц ISM



радіодіапазоні. Ця технологія швидко розвивається Wi-Fi Alliance на базі такого стандарту як IEEE (Institute of Electrical and Electronics Engineers) 802.11. Ця технологія де-факто стала практично повсюдною, в світі випускаються мільярди Wi-Fi пристроїв на рік. Wi-Fi спочатку розроблялася для локальної зв'язку. Сучасні точки доступу зі шток-антенами можуть забезпечити радіус дії приблизно до 100 м без перешкод. Існують рішення з використанням підсилювача і напівпараболічної антени з дальністю понад 20 км [5].

Ця технологія не стоїть на місці, постійно розвиваючись. Так, технологія Wi-Fi Direct дозволяє пристроям Wi-Fi підключатися безпосередньо без точки доступу та мережі. Пристроєм вдається з'єднатися одночасно один з одним або з групою з декількох пристроїв. Підключаються Wi-Fi Direct-сертифіковані пристрої легко і просто: або два NFC-сумісних пристроїв разом, або з введенням ПІН-коду. Крім того, всі прямі з'єднання Wi-Fi захищені WPA2. Wi-Fi з'єднання можуть бути легко порушені або швидкість з'єднання знижена при наявності в тій же області інших подібних пристроїв [5].

#### Переваги Wi-Fi:

- поширення в світі є повсюдне;
- сумісність є гарантованою;
- надзвичайно висока швидкість для передачі даних;
- надійність є високою.

#### Недоліки Wi-Fi:

- перешкоди та інтерференція;
- з безпекою деякі проблеми;
- проникаюча здатність є невисока в міській забудові;
- енергоємність є високою;
- діапазон та обмеження в різних країнах є різні [5].

### 1.1.13 Wi-Fi HaLow

Нещодавно на CES 2016 Wi-Fi Alliance анонсував розробку нового стандарту бездротового зв'язку, призначеного для роботи Інтернету речей. Новий стандарт отримав назву HaLow і поки не затверджену специфікацію IEEE 802.11ah. Сертифікація перших пристроїв, сумісних з Wi-Fi HaLow, почнеться в 2018 році, проте продукти з підтримкою нової специфікації з'являться на ринку раніше. Для підключення Wi-Fi HaLow буде використовуватися неліцензована частота 900 МГц. Це помітно збільшить проникаючу здатність сигналу в міській забудові, а радіус її дії буде набагато більше, ніж у сучасного бездротового стандарту, – до 1 кілометра. У той же час платою за «далекобійність» є мала потужність сигналу. Пропускна здатність Wi-Fi HaLow буде набагато нижче, ніж максимум Wi-Fi 802.11ac (7 Гбіт / с), передбачувана швидкість: 50 кбіт / с – 18 Мбіт / с. За заявою альянсу, HaLow буде широко використовувати існуючі протоколи Wi-Fi, що забезпечить високий рівень сумісності і безпеки [5].

### 1.1.14 Thread

Метою Thread Group є розроблення найкращого способу управління та підключення пристроїв в будинку. Доповнення до Wi-Fi є Thread, який має для використання в домашньої автоматизації в плані енергоспоживання та безпеки чіткі обмеження. Саме на стандарті 6LoWPAN заснований протокол. Цей протокол забезпечує надійну безпеку банківського класу AES на додаток до надійності меш-мережі, яка була розроблена спеціально для домашньої автоматизації. Можна під'єднати 250 авторизованих пристроїв до однієї мережі [5].

Переваги Thread:

- добре доповнення до Wi-Fi;
- для домашньої електроніки спеціальна розробка;
- надійна система;
- використання відкритих стандартів, які перевірені;

- безпека є високою;
- енергоспоживання низьке.

Недоліки Thread:

- перешкоди та інтерференція;
- в міській забудові невисока проникаюча здатність;
- в різних країнах обмеження та діапазон різні [5].

#### 1.1.15 ZigBee

Специфікацією мережевих протоколів верхнього рівня називають ZigBee, який регламентований стандартом IEEE 802.15.4. З'явився він у 2003 році. Описують ZigBee й IEEE 802.15.4 персональні бездротові обчислювальні мережі (WPAN, wireless personal area networks). ZigBee специфікація орієнтована на програми вимагають гарантованої безпечної передачі даних при можливості тривалої роботи мережевих пристроїв від автономних джерел живлення і невеликих швидкостях. ZigBee технологія підтримує топології мережі («точка–точка», «зірка» та «дерево»), а також самовідновлюється та самоорганізується порожнисту (mesh) топологію з маршрутизацією й ретрансляцією повідомлень [5].

Переваги ZigBee:

- здатність до самовідновлення та самоорганізації ;
- є простота розгортання;
- стійкість є високою;
- відкритий стандарт;
- надзвичайно висока безпека;
- формуйте сітчасту мережу;
- енергоспоживання є низьке.

Недоліки ZigBee:

- швидкість невисока;

- дуже велика частина трафіку на передачу пакетів витрачається, які містять інформацію по синхронізації і адресну інформацію і т.п.;
- проникаюча здатність невисока в міській забудові;
- рівень стандартизації недостатньо високий та відсутність єдиної програмно-апаратної платформи для розробки складних додатків [5].

#### 1.1.16 Z-Wave

Запатентований бездротовий протокол зв'язку називають Z-Wave, в основному для домашньої автоматизації розроблений. Ця технологія використовує мініатюрні та малопотужні модулі, а саме радіочастотні, які вбудовуються в різні пристрої та побутову електроніку. Z-Wave оптимізований для передачі не складних керуючих команд з дуже малими затримками і працює в діапазоні частот до 1 ГГц. В основі Z-Wave є система, що самоорганізується порожниста мережа (mesh мережу), в якій кожен пристрій або вузол може передавати та приймати сигнали інших пристроїв мережі, використовуючи сусідні проміжні вузли [5].

Переваги Z-Wave:

- гарантована сумісність Z-Wave;
- формуєте сітчасту мережу ;
- мало живиться і може працювати від акумулятора;
- мережеві мережі стають надійнішими завдяки більшій кількості пристроїв;
- використовує 0,9 ГГц і може передавати до 100 метрів;
- найменш вірогідна з цих трьох перешкода сигналу від твердих предметів або зовнішніх джерел [5].

Недоліки Z-Wave:

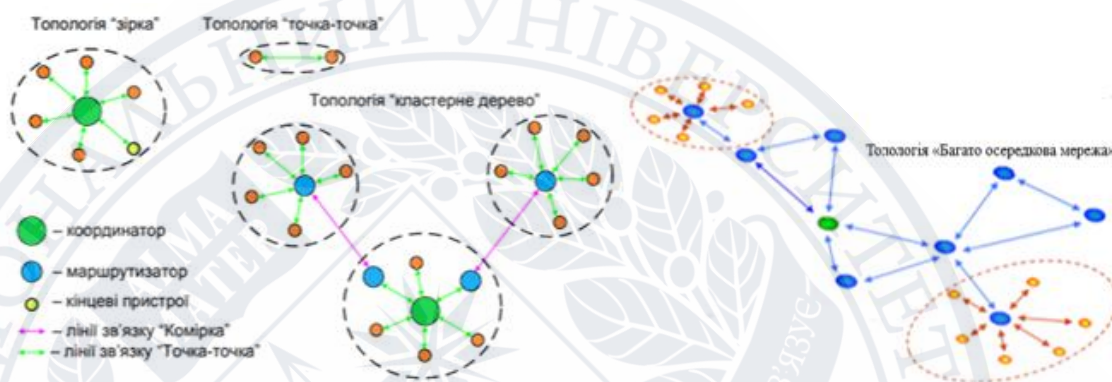
- закритий протокол;
- витрачає найбільше;
- максимум чотири стрибки в сітці;



- може підтримувати до 230 пристроїв в мережі;
- використовує 0,9 ГГц, що є найповільнішим з усіх протоколів [5].

## 1.2 Топології бездротових мереж

Працюють перераховані бездротові мережі в одному або декількох варіантах топології. Наведені топології бездротових мереж різних конфігурацій на рис. 1 [6].



**Рисунок 1.1** – Топологія бездротових мереж

### 1.2.1 Топологія точка–точка

Організації мережі з двох пристроїв найпростіший варіант. Вузли цієї мережі, як правило є рівноправними, тобто мережа однорангова. Топологія є характерною для ANT, Bluetooth, RuBee, RFID, WI-FI, PDC, UWB, WI-FI, ZigBee, Insteon і інших [6].

### 1.2.2 Топологія «Зірка»

Топологія «Зірка» для всіх сучасних мереж зв'язку та обчислювальних мереж є основою організації. Використовують представлену топологію протоколи ZigBee, WI-FI, Insteon, CDMAOne, UWB, GSM, IDEN, UTMS, WIMAX, GPRS, [6].

### 1.2.3 Топологія «Багато осередкова мережа»

Багато осередкова мережа – базова повністю зв'язана топологія мереж зв'язку та комп'ютерних мереж. З усіма іншими робочими станціями цієї мережі з'єднується кожна робоча станція мережі. Характеризується вона складністю налаштування, високою стійкістю до відмов та надмірною витратою кабелю в провідних мережах. Кілька можливих шляхів сполучення з іншими вузлами має кожен вузол, тому за рахунок цього така топологія дуже стійка. Не призводить до втрати з'єднання між двома комп'ютерами, зникнення одного з каналів. Ця топологія характерна, як правило для великих мереж. Допускає з'єднання великої кількості вузлів та будується з повним-шляхом видалення деяких зв'язків. Застосована топологія може бути для мереж з використанням протоколів ZigBee, WIMAX, UWB, UTMS, WI-FI, CDMAOne, Insteon, UWB, IDEN, GSM, GPRS [6].

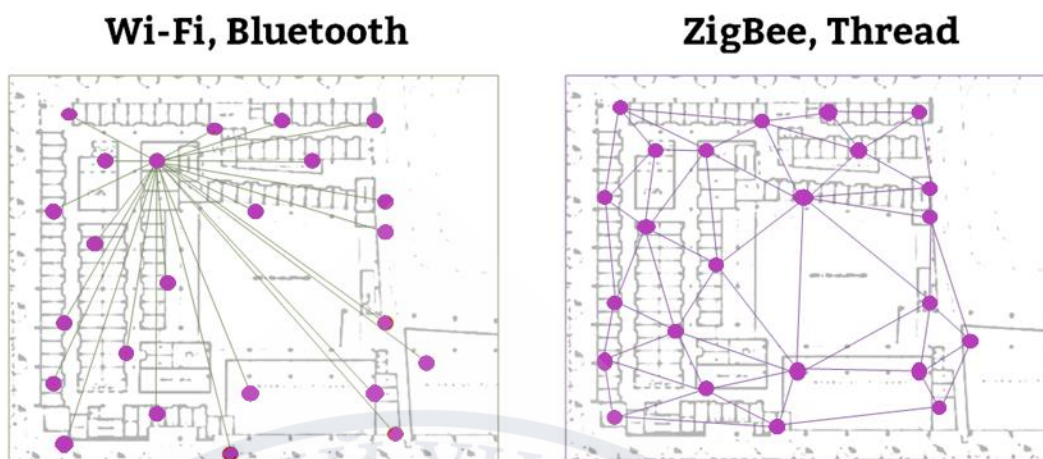
### 1.2.4 Топологія «кластерне дерево»

Утворюється топологія «Кластерний дерево» у вигляді комбінацій вищезгаданих топологій обчислювальних мереж. В точці (корінь) розташовується підстава дерева обчислювальної мережі, в якій збираються комунікаційні лінії інформації (гілки дерева). Де неможливо безпосереднє застосування базових мережних структур в чистому вигляді, будуються обчислювальні мережі з деревоподібної структурою [6].

Всі перераховані вище технології здатні працювати за схемою «точка–точка» [7]. Якщо потрібно створити мережу датчиків, то логічно використовувати топологію типу «зірка» (наприклад, WI-FI) [6].

## 1.3 Порівняння мережевих топологій

Про підтримувані мережеві топології в попередньому розділі було сказано, але не було сказано про особливості. Розглянемо такий приклад (рис 1.2) [7]:

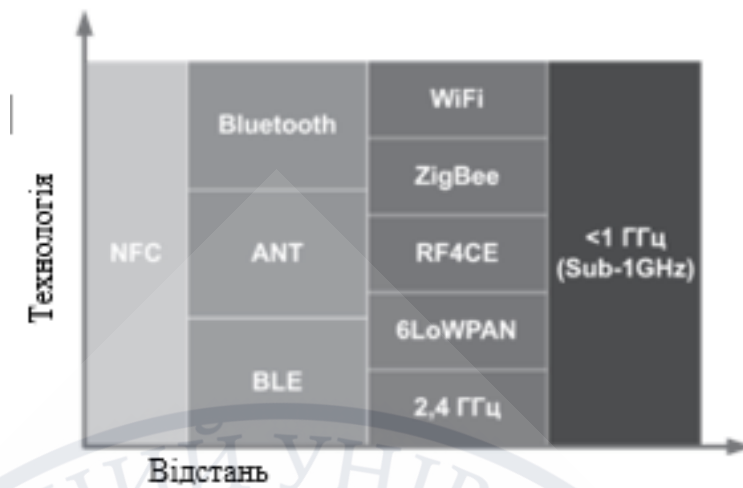


**Рисунок 1.2 – «Зірка» vs. «Mesh»**

Мережева взаємодія йде в мережах Bluetooth і Wi-Fi через центральний шлюз. Обмін даними стане неможливим, якщо він вийде з ладу. Окремі вузли можуть залишитися без зв'язку, якщо несподівано виникла перешкода на шляху проходження радіосигналу. Надійність зв'язку у мережах ZigBee і Thread підвищується за рахунок наявності надлишкових зв'язків між пристроями. Всі пристрої, які не йдуть в сплячий режим виконують роль роутерів. ZigBee-мережа буде продовжувати функціонувати далі, якщо з ладу вийде пристрій, який виступав в якості організатора мережі. За рахунок наявності надлишкових зв'язків виникнення перешкоди, а також вихід будь-якого з роутерів з ладу не є критичним. З введенням додаткових вузлів, які можуть виконувати завдання роутера і мають стаціонарне харчування, мережа стає надійніша [7].

Значна дальність дії субгігагерцових рішень (рис. 1.3) позначається на швидкості передачі даних [7] (рис. 1.4). Найбільшою пропускнуою спроможністю володіє Wi-Fi, який, разом з тим, відрізняється і високим споживанням [7].



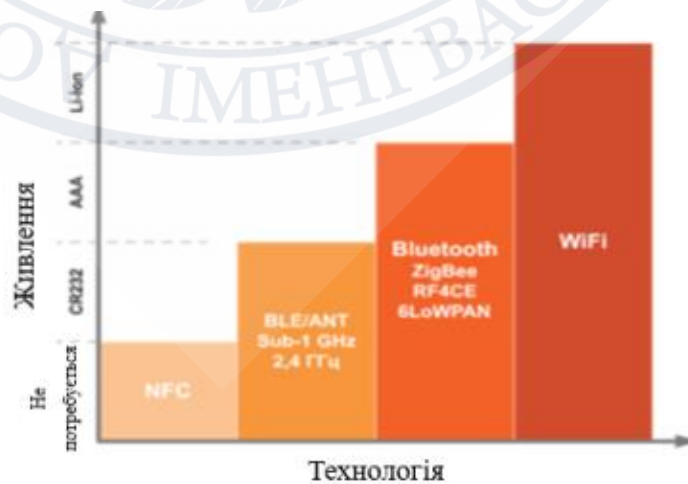


**Рисунок 1.3** – Порівняння дальності дії сучасних бездротових технологій



**Рисунок 1.4** – Пропускна здатність сучасних бездротових технологій

NFC програє всім іншим технологіям за швидкістю і дальністю дії, але бере впевнений реванш за рівнем споживання [7], а також безпекою даних, завдяки дуже малій відстані зв'язку (рис. 1.5) [7].



**Рисунов 1.5** – Споживання сучасних бездротових рішень



NFC–транспондери можуть і зовсім обходитися без власного елемента живлення, використовуючи для роботи енергію радіовипромінювання ведучого.

При виборі оптимальної технології необхідно враховувати і таку особливість, як складність реалізації. Наприклад, якщо мова йде про створення захищеного Wi-Fi–пристрої, то розробники повинні мати досить високу кваліфікацію. Це відноситься як до схемотехніки, так і до програмістам, на плечі яких ляже написання відповідних протоколів. Будемо відверті – далеко не всім під силу впоратися з цими завданнями. Наявність на ринку готових рішень виявляється справжньою паличкою–виручалочкою для невеликих компаній. Наприклад, мікроконтролери з бездротовими модулями виробництва компанії Texas Instruments не тільки спрощують створення принципової схеми і розведення друкованої плати, але і мають підтримку у вигляді налагоджувальних наборів, програмних бібліотек і готових стеків популярних протоколів (ZigBee, Bluetooth і так далі) [8].

#### **1.4 Типова структура мережі Wi-Fi**

В якості бездротової технології передачі даних в складі системи обрана технологія Wi-Fi. Корпорацією NCR / AT & T в Нідерландах в 1991 році був винайдений [9].

Можна за допомогою цієї технології обмінюватися інформацією між двома або більше пристроями. Розроблений Wi-Fi для мобільних обчислювальних пристроїв, таких як ноутбуки, але зараз для мобільних додатків та побутової електроніки він широко використовується, таких як телевізори, DVD–програвачі та цифрові камери. Повинно бути дві можливості у спілкуванні із з'єднанням Wi-Fi, які можуть здійснюватися через точку доступу до підключення клієнта або від підключення клієнта до клієнта. Wi-Fi зазвичай називають бездротовою локальною мережею. Wi-Fi технологія дозволяє локальним мережам працювати без кабелю та проводки. Для домашніх та ділових мереж – це робить популярний вибір. Бездротовий адаптер комп'ютера передає дані в радіосигнал і передає дані в антену для користувачів [9].

Wi-Fi використовує сімейство стандартів IEEE 802.11. Багато версій є – 802.11a, 802.11b тощо. Типова бездротова точка доступу, що використовує 802.11b або 802.11g зі стандартною антеною, може мати дальність дії 32 м (120 футів) у приміщенні та 95 м (300 футів) на відкритому повітрі [10].

Однак новіший IEEE 802.11n може перевищувати цей діапазон більш ніж у два рази ( Wikipedea про Wi-Fi). Мережі 802.11b або 802.11g використовують несучу частоту 2,5 ГГц, тоді як новіші мережі 802.11n використовують частоту 5 ГГц. Пристрої, що підтримують обидві частоти ( 2,4 і 5 ГГц ), відомі як двосмугові , і корисні при переході від старих бездротових мереж до нових мереж [10].

Стандарти Wi-Fi зворотно сумісні, тому, якщо ви купуєте пристрій, який використовує найновіший стандарт Wi-Fi, він буде працювати в мережах, які підтримують лише старіші стандарти. У 2018 році альянс Wi-Fi запровадив простішу схему іменування, і 802.11n тепер відомий як Wi-Fi 4. Крім того, новіші пристрої також відображатимуть це число разом із силою сигналу [10].

Wi-Fi – це недорога технологія бездротового зв'язку. Налаштування Wi-Fi складається з бездротового маршрутизатора, який обслуговує центр зв'язку, що пов'язує портативний пристрій з підключенням до Інтернету. Ця мережа полегшує підключення багатьох пристроїв залежно від конфігурації маршрутизатора. Діапазон цих мереж обмежений через низьку потужність передачі, що дозволяє користувачеві підключатися лише в безпосередній близькості [11].

Wi-Fi – це форма низькопотужного бездротового зв'язку, що використовується багатьма електронними пристроями, такими як ноутбуки, системи, смартфони тощо. У налаштуваннях Wi-Fi бездротовий маршрутизатор служить центром зв'язку. Діапазон цих мереж надзвичайно обмежений через низьку потужність передач, що дозволяє користувачам підключатися лише в безпосередній близькості від маршрутизатора або повторювача сигналу. Wi-Fi є загальним у домашніх мережевих програмах, що забезпечує портативність без

потреби в кабелях. Мережі Wi-Fi повинні бути захищені паролями з метою безпеки, щоб інші користувачі не мали до них доступу [11].

Ця мережа полегшує підключення багатьох пристроїв залежно від конфігурації маршрутизатора. Діапазон цих мереж обмежений через низьку потужність передачі, що дозволяє користувачеві підключатися лише в безпосередній близькості [11].

Переваги:

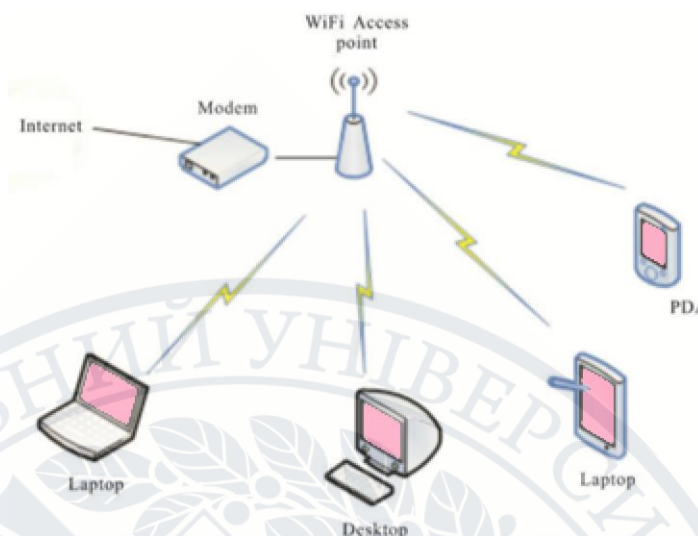
- Простота інтеграції та зручність – бездротовий характер таких мереж дозволяє користувачам отримувати доступ до мережевих ресурсів майже з будь-якого зручного місця [11];
- Мобільність – З появою загальнодоступних бездротових мереж, отримати доступ до Інтернету користувачі можуть навіть за межами свого звичайного робочого середовища [11];
- Розширюваність – бездротові мережі здатні обслуговувати раптово зростаючу кількість клієнтів за допомогою існуючого обладнання. У дротовій мережі додаткові клієнти потребують додаткового підключення [11].

Недоліки:

- Бездротові локальні мережі можуть бути небажаними з ряду причин [11];
- Радіочастотні передачі та бездротові мережеві сигнали піддаються найрізноманітнішим перешкодам, включаючи складні ефекти розповсюдження, які поза контролем адміністратора мережі [11];
- Проблеми з безпекою – бездротові мережі можуть вибрати деякі з різних технологій шифрування [11];
- Діапазон буде недостатнім для більшої конструкції – і, щоб збільшити її діапазон, потрібно придбати ретранслятори або додаткові точки доступу [11];
- Швидкість у більшості бездротових мереж буде повільнішою, ніж у найповільніших звичайних дротових мереж [11];



- Встановлення інфраструктурної бездротової мережі є складним для налаштування [11].



**Рисунок 1.6 – З'єднання з Wi-Fi**

Елементи Wi-Fi включають наступне:

- Точка бездротового доступу – використовується, щоб дозволити бездротовим пристроям підключатися до бездротової мережі. Така бездротова мережа, як Cisco, полегшить роботу нових пристроїв, підключивши їх до Інтернету та надаючи хорошу підтримку мобільним користувачам. Функція бездротової точки доступу в мережі подібна до функції підсилювача в домашній стереосистемі. Точка доступу використовує пропускну здатність для розтягування, щоб кілька пристроїв могли їздити по мережі з далекої відстані. Однак точка бездротового доступу дає дуже корисні дані про пристрої через мережу для забезпечення безпеки, а також для практичних цілей [9].
- Карти Wi-Fi. Ці картки передають бездротовий сигнал, а також інформацію реле, яка може бути внутрішньою або зовнішньою. Карти Wi-Fi також відомі як адаптери [9].
- Запобіжні заходи. Брандмауери, а також захищають такі мережі, як антивірусне програмне забезпечення, від непрошених користувачів та захищають інформацію [9].

З точки зору розв'язуваної задачі в даній системі застосування технології Wi-Fi оптимально. Тому що вона виявилася легше і з нею краще працювати.



## РОЗДІЛ 2. МОДУЛЬ ПЕРЕДАЧІ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ Wi-Fi

### 2.1 Побудова каналу зв'язку за допомогою мікроконтролера ESP32

При проектуванні роботи було проведено та відібрано тести багатьох схем підключення модулів. Кожний з вибраних варіантів мав свої переваги та недоліки, але необхідно було вибрати найкращий варіант не зважаючи на її недосконалість. Вирішальним питанням стали дешевизна, максимально зручне підключення та можливість «гарячої» заміни модулів. Це дозволяє швидко замінювати модулі, які було вирішено замінити на кращі або якщо якийсь пристрій вийшов з ладу, потрібно його замінити [12].

Кілька років тому ESP8266 взяв вбудований світ IoT штурмом. Менше ніж за 3 долари ви можете отримати програмований мікроконтролер з підтримкою Wi-Fi, який може відслідковувати роботу з будь-якої точки світу [12]. Зараз Espressif (напівпровідникова компанія, що стоїть за ESP8266) випустив переобтяжене оновлення [12] : ESP32 [13].

Будучи наступником ESP8266; він не тільки підтримує Wi-Fi, але також має Bluetooth 4.0 (BLE / Bluetooth Smart) – ідеально підходить для будь-якого проекту IoT [12].

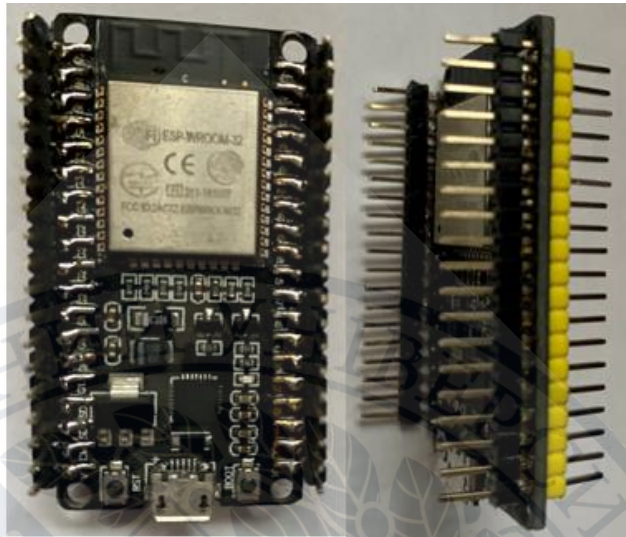
Плата розробників оснащує модуль ESP-WROOM-32, що містить двоядерний 32-розрядний мікропроцесор LX6 Tensilica Xtensa® [14].

Цей процесор схожий на ESP8266, але має два ядра центрального процесора (можна керувати індивідуально), працює на частоті від 80 до 240 МГц та працює з частотою до 600 DMIPS (інструкція Dhrystone Million Per Second) [12].

Чіп ESP-WROOM-32:

- Xtensa® двоядерний 32-розрядний LX6;
- Частота тактової частоти до 240 МГц;
- 520 Кб внутрішньої SRAM;
- Зовнішня пам'ять 4 Мб;

- 802.11b / g / n трансивер Wi-Fi;
- Bluetooth 4.2 / BLE [12].



**Рисунок 2.1 – ESP-WROOM-32**

Також є 448 КБ ПЗУ, 520 КБ SRAM і 4 Мб флеш-пам'яті (для зберігання програм та даних), достатньо для того, щоб впоратися з великими ланцюжками, що складаються з веб-сторінок, JSON / XML-даними та усім, що ми сьогодні кидаємо на пристрої IoT [12].

ESP32 інтегрує 802.11 b/g /n HT40 Wi-Fi трансивер, тому він може не тільки підключатися до мережі Wi-Fi та взаємодіяти з Інтернетом, але також може створити власну мережу, що дозволяє іншим пристроям підключатися безпосередньо до це. ESP32 також підтримує Wi-Fi Direct , що є хорошим варіантом для однорангового з'єднання без необхідності точки доступу. Wi-Fi Direct простіше в налаштуванні і швидкості передачі даних набагато краще , ніж Bluetooth. Чіп також має подвійний режим Bluetooth, тобто він підтримує як Bluetooth 4.0 (BLE / Bluetooth Smart), так і Bluetooth Classic (BT) , що робить його ще більш універсальним [12].

Оскільки діапазон робочої напруги ESP32 становить від 2,2 до 3,6 В , плата постачається з регулятором напруги LDO, щоб підтримувати стабільну напругу на рівні 3,3 В. Він може надійно подавати до 600 мА , що має бути більш ніж достатньо, коли ESP32 підтягує до 250 мА під час радіочастотних передач . Вихід регулятора також розбивається на одну зі сторін плати і позначається як 3V3 [12].

Вимоги до живлення:

- Робоча напруга: від 2,2 до 3,6 В;
- Вбудований регулятор 3,3 В 600 мА;
- 5 мкА в режимі сну;
- 250 мА під час радіочастотних передач [12].

Живлення плати розробки ESP32 подається через вбудований USB-роз'єм MicroB. Крім того, за наявності джерела регульованого 5V напруги контакт VIN може використовуватися для безпосереднього живлення ESP32 і його периферійних пристроїв [12].

Також струм сну мікросхеми ESP32 становить менше 5 мкА, що робить його придатним для застосування в електроніці з живленням від акумулятора і електронікою, що носить [12].

Хоча ESP32 має в цілому 48 контактів GPIO, тільки 25 з них розбиті на конектори контактів на обох сторонах плати розробки. Ці контакти можуть бути призначені для всіх типів периферійних пристроїв, включаючи [12]:

- 15 каналів АЦП – 15 каналів 12-бітових АЦП SAR. Діапазон АЦП можна встановити в мікропрограмі на 0–1В, 0–1,4В, 0–2В або 0–4В [12];
- 2 інтерфейси UART – 2 інтерфейси UART. Один використовується для послідовного завантаження коду. Вони оснащені контролем потоку та також підтримують IrDA [12];
- 25 ШІМ-виходів – 25 каналів ШІМ-контактів для затемнення світлодіодів або управління двигунами [12];
- 2 ЦАП-канали – 8-бітові ЦАП для отримання аналогових напружень [12];
- Інтерфейси SPI, I2C та I2S – є 3 інтерфейси SPI та 1 I2C для підключення всіляких датчиків та периферійних пристроїв, а також два інтерфейси I2S, якщо ви хочете додати звук до свого проекту [12];
- 9 сенсорних панелей – 9 GPIO мають ємнісне зондування [12].

Мультиплексовані вводи-виводи [12]:



- 15 каналів АЦП;
- 2 інтерфейси UART;
- 25 ШІМ–виходів;
- 2 канали ЦАП;
- Інтерфейс SPI, I2C та I2S;
- 9 Сенсорні панелі [12].

Завдяки функції мультиплексування контактів ESP32 (мультиплексування декількох периферійних пристроїв на одному контакті GPIO). Це означає, що один контакт GPIO може працювати як вхід/вихід ЦАП/сенсорна панель АЦП [12].

На платі розробки ESP32 є дві кнопки. Кнопка Reset позначена як EN, розташована у верхньому лівому куті, яка, звичайно, використовується для скидання чіпа ESP32. Інша кнопка завантаження в нижньому лівому куті – це кнопка завантаження, яка використовується під час завантаження нового ескізу / програм [12].

Перемикачі та індикатори [12].:

- UK – скидання чіпа ESP32;
- Завантаження – Завантажте нові програми;
- Червоний світлодіод – індикатор живлення;
- Синій світлодіод – програмований користувачем [12].

Плата включає CP2102 USB-to-UART Bridge Controller від Silicon Labs , який легко перетворює USB-сигнал у послідовний і дозволяє вашому комп'ютеру програмувати та взаємодіяти з чіпом ESP32 [12].

Послідовний зв'язок:

- CP2102 USB-UART-перетворювач;
- Швидкість зв'язку 5 Мбіт / с;
- Підтримка IrDA [12].

Плата розробки ESP32 має загалом 30 контактів, які взаємодіють із зовнішнім світом. З'єднання такі [12]:



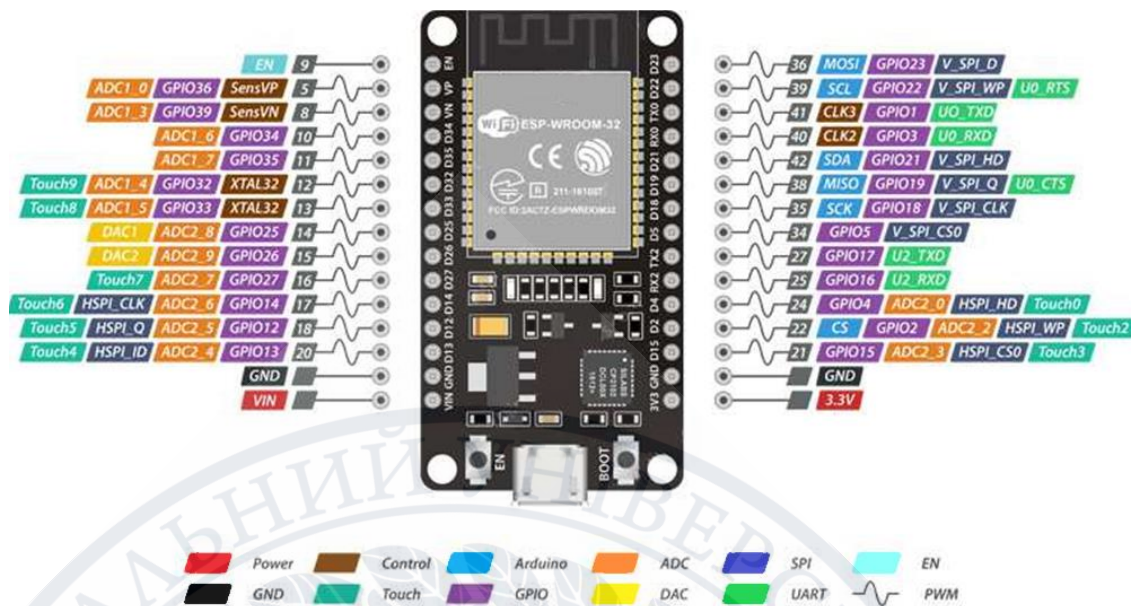


Рисунок 2.2 – ESP–WROOM–32 [12].

Для простоти ми створимо групи контактів з аналогічними функціональними можливостями [12].

— Контакт живлення. Є два контакти живлення, а саме: VIN–контакт і контакт 3,3 V. Контакт VIN можна використовувати для безпосереднього живлення ESP32 та його периферійних пристроїв, якщо у вас є регульоване джерело напруги 5 В. Контакт 3,3 В є виходом вбудованого регулятора напруги. Цей контакт можна використовувати для живлення зовнішніх компонентів [12].

— GND - контакт заземлення ESP32 плати розробки [12].

— Контакти Arduino - це не що інше, як апаратні I2C ESP32 і контакти SPI для підключення всіляких датчиків і периферійних пристроїв у проекті [12].

— Контакти GPIO. Плата розробки ESP32 має 25 контактів GPIO, які програмно можуть бути призначені для різних функцій. Кожен GPIO з цифровою підтримкою може бути сконфігурований для внутрішнього підйому або опускання, або може бути встановлений на високий опір. Під час налаштування як вхідні дані він також може бути встановлений у стан edge-trigger або level-trigger для генерації переривань CPU [12].

- Канали ADC. Плата інтегрує 12-бітові АЦП SAR і підтримує вимірювання на 15 каналах (аналогові висновки) [12].
- Канали ЦАП. Плата має два 8-бітових ЦАП-канали для перетворення цифрових сигналів у аналогові напруги. Цей подвійний ЦАП може керувати іншими схемами [12].
- Сенсорні панелі. Плата пропонує 9 ємнісних зондуючих GPIO, які виявляють ємнісні зміни, спричинені прямим контактом GPIO або безпосередньою близькістю до пальця або інших предметів [12].
- Канали UART. Плата розробки ESP32 має 2 інтерфейси UART, тобто UART0 та UART2, які забезпечують асинхронний зв'язок (RS232 та RS485) та підтримку IrDA, і здійснюють зв'язок зі швидкістю до 5 Мбіт / с. UART також забезпечує апаратне управління сигналами CTS та RTS, а також програмне управління потоком (XON та XOFF) [12].
- Канали SPI. Канали SPI ESP32 оснащені трьома SPI (SPI, HSPI та VSPI) у ведучому та веденому режимах. SPI підтримують наступні функції SPI загального призначення [12]:
  - 4 режими синхронізації передачі формату SPI;
  - До 80 МГц та розділені тактові частоти 80 МГц;
  - До 64-байтового FIFO [12].Усі SPI також можна використовувати для підключення до зовнішнього виблискування / SRAM та РК-дисплея [12].
- ШІМ-канали. Плата має 25 каналів (Майже всі виводи GPIO) ШІМ-каналів, керованих контролером модуляції широти імпульсу (ШІМ). ШІМ-вихід можна використовувати для керування цифровими двигунами та світлодіодами. Контролер складається з ШІМ-таймерів та ШІМ-оператора. Кожен таймер забезпечує синхронізацію в синхронній або незалежній формі, і кожен оператор ШІМ генерує форму сигналу для одного каналу ШІМ [12].

– EN Pin використовується для включення ESP32. Мікросхема вмикається при натисканні HIGH. При натисканні LOW мікросхема працює при мінімальній потужності [12].

## 2.2 Трьохосьовий гіроскоп з трьохосьовим акселерометром MPU6050

Модуль датчика MPU6050 є повноцінним 6-осьовим пристроєм відстеження руху. Датчик поєднує в собі 3-осьовий акселерометр, 3-осьовий гіроскоп і цифровий процесор руху – все це в невеликій упаковці. Крім того, він має додаткову функцію вбудованого датчика температури. Він має інтерфейс шини I2C для зв'язку з мікроконтролерами. Він має допоміжну шину I2C для зв'язку з іншими датчиками, такими як 3-осьовий магнітометр, датчик тиску тощо. Якщо 3-осьовий магнітометр підключений до допоміжної шини I2C, то MPU6050 може забезпечити повний 9-осьовий вихід Motion Fusion [15].

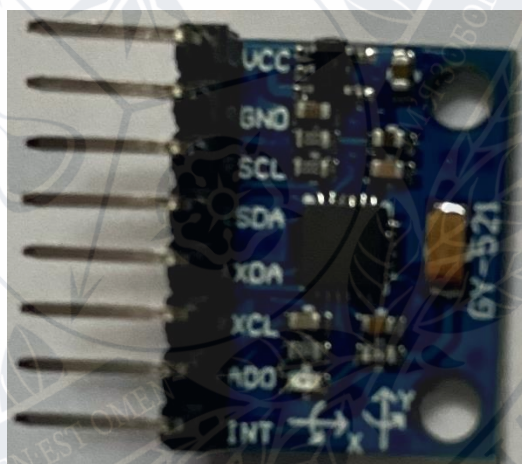


Рисунок 2.3 – Датчик MPU6050

MPU6050 складається з 3-осьового гіроскопа з технологією Micro Electro Mechanical System (MEMS). Він використовується для виявлення швидкості обертання вздовж осей X, Y, Z, як показано на малюнку нижче [15].

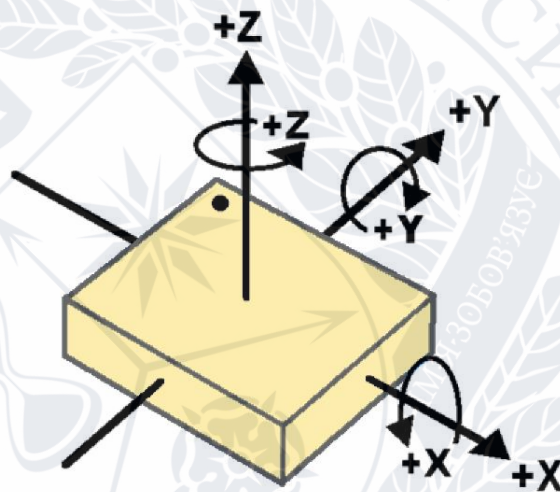
Гіроскоп вимірює швидкість обертання (рад / с) - це зміна кутового положення з плином часу вздовж осей X, Y та Z (крен, крок і похибка). Це дозволяє нам визначити орієнтацію об'єкта [16].



Акселерометр вимірює прискорення (швидкість зміни швидкості руху об'єкта). Він відчуває статичні фокуси, такі як гравітація ( $9,8 \text{ м / с}^2$ ) або динамічні сили, такі як вібрації або рух. MPU-6050 вимірює прискорення по осі X, Y та Z. В ідеалі в статичному об'єкті прискорення по осі Z дорівнює силі тяжіння, і воно повинно бути нулем на осях X і Y [16].

Використовуючи значення акселерометра, можна розрахувати кути нахилу та кроку за допомогою тригонометрії, але неможливо розрахувати гойдалку.

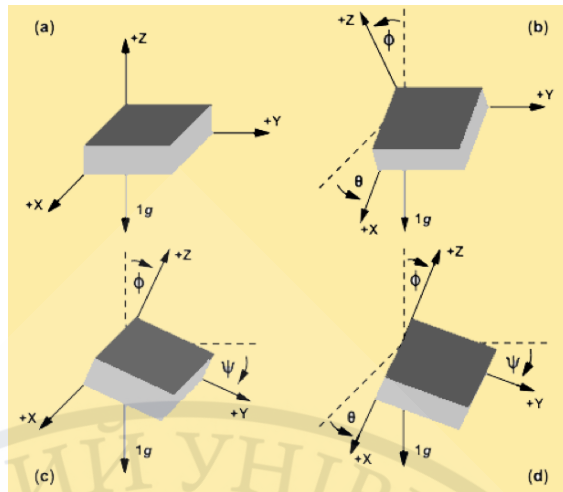
Ми можемо поєднати інформацію від обох датчиків, щоб отримати точну інформацію про орієнтацію датчика [16].



**Рисунок 2.4** – 3-осьовий гіроскоп MPU6050

- Коли гіроскопи обертаються навколо будь-якої з сенсорних осей, ефект Коріоліса викликає вібрацію, яка виявляється MEM всередині MPU6050 [15].
- Отриманий сигнал посилюється, демодулюється та фільтрується, щоб отримати напругу, пропорційну кутовій швидкості [15].
- Ця напруга оцифровується за допомогою 16-бітового АЦП для вибірки кожної осі [15].
- Повномасштабний діапазон виходу становить  $\pm 250 \text{ мВ}$ ,  $\pm 500 \text{ мВ}$ ,  $\pm 1000 \text{ мВ}$ ,  $\pm 2000 \text{ мВ}$  [15].
- Він вимірює кутову швидкість вздовж кожної осі в градусах на секунду [15].





**Рисунок 2.5** – 3-осьовий гіроскоп MPU6050

- Прискорення вздовж осей відхиляє рухому масу [15].
- Це зміщення рухомої пластини (маси) врівноважує диференціальний конденсатор, що призводить до виходу датчика. Вихідна амплітуда пропорційна прискоренню [15].
- 16-розрядний АЦП використовується для отримання оцифрованого виводу [15].
- Повномасштабний діапазон прискорення становить  $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$ ,  $\pm 16g$  [15].
- Вимірюється в Н (ньютон) (одиниця сили тяжіння) [15].
- Коли пристрій розміщений на рівній поверхні, він буде вимірювати  $0g$  на осі X та Y та  $+1g$  на осі Z [15].

Вбудований цифровий процесор руху (DMP) використовується для обчислення алгоритмів обробки руху. Він бере дані з гіроскопа, акселерометра та додаткових сторонніх датчиків, таких як магнітометр, і обробляє дані. Він забезпечує дані руху, такі як нахил, крок, кути похилу, ландшафт і портрет, і т.д. Це мінімізує процеси хоста при обчисленні даних руху. Отримані дані можна прочитати з реєстрів DMP [15].

Вихід датчика температури мікросхеми оцифровується за допомогою АЦП. Показання з датчика температури можна прочитати з реєстру даних датчика [15].

Модуль MPU-6050 має 8 контактів [15]:

INT: Переривання виводу цифрового сигналу.

AD0: LSB-адреса підлеглої адреси I2C. Це 0-й біт у 7-розрядній адресі підлеглого пристрою. При підключенні до VCC він зчитується як логічний один, і адреса підлеглих змінюється.

XCL: вивід додаткового послідовного тактового генератора. Цей контакт використовується для з'єднання інших сенсорів SCL з підтримкою інтерфейсу I2C до MPU-6050.

XDA: вивід допоміжних послідовних даних. Цей контакт використовується для підключення інших сенсорів SDA з підтримкою інтерфейсу I2C до MPU-6050.

SCL: контакт послідовного тактового генератора. Підключіть цей контакт до контакту SCL мікроконтролерів.

SDA: вивід послідовних даних. Підключіть цей контакт до контакту SDA мікроконтролерів.

GND: контакт заземлення. Підключіть цей контакт до заземлення.

VCC: контакт живлення. Підключіть цей контакт до постачання постійного струму + 5 В.

Модуль MPU-6050 має підлеглу адресу (коли AD0 = 0, тобто він не з'єднаний з Vcc), оскільки [15]: Адреса запису підлеглих (SLA + W) : 0xD0; Адреса зчитування підлеглих (SLA + R) : 0xD1 [15].

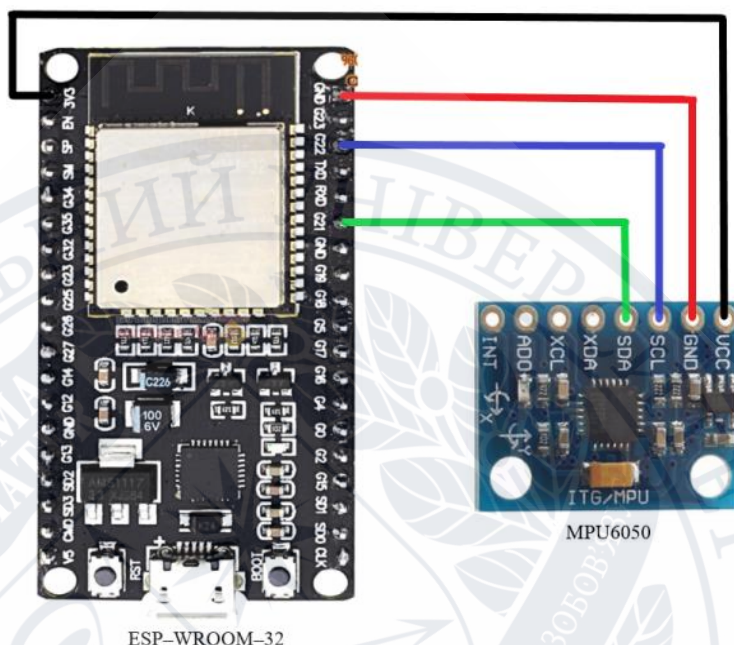
### **2.3 Модуль GY-521, трьохосьовий гіроскоп з трьохосьовим акселерометром MPU6050 на базі мікроконтролера ESP32**

Сенсорний модуль MPU6050 складається з акселерометра і гіроскопа в одній мікросхемі. Він містить 16 біт АЦП для кожного каналу. Таким чином, він може зчитувати дані по каналах осей x, y і z одночасно. Датчик використовує I2C-шину для взаємодії з Arduino або будь-яким іншим мікро контролером [17].

Сенсорний модуль MPU-6050 використовується в багатьох додатках, таких як дрони, роботи, датчики руху. В роботі використано акселерометр і

гіроскоп MPU6050 з мікроконтролером ESP32, який виводить значення про прискорення і кутової швидкості через веб-сервер [17].

Для початку підключимо MPU6050 до ESP32. Принципова схема підключення акселерометра / гіроскопа MPU6050 до ESP32 приведена далі [17].



**Рисунок 2.6** – Під'єднання датчика MPU6050 до мікроконтролера ESP-WROOM-32

MPU6050 зв'язується з ESP32 по протоколу I2C, тому нам потрібно чотири дроти для взаємодії ESP32 і MPU6050. Як показано на рисунку вище, підключено лінію SDA MPU6050 до контакту G21 на ESP32, лінію SCL до контакту G22, GND до GND і VCC до контакту 3V3. Інші штирі MPU6050 не потрібно підключати [17].

MPU6050 заснований на технології мікромеханічних систем (MEMS). Цей датчик має 3-осьовий гіроскоп, 3-осьовий акселерометр та вбудований датчик температури. Його можна використовувати для вимірювання таких параметрів, як прискорення, швидкість, орієнтація, зміщення і т. Д. Модуль MPU6050 має невеликі розміри і низьке енергоспоживання, високу повторюваність, високу стійкість до ударів і низьку ціну. MPU6050 поставляється з шиною I2C і може легко взаємодіяти з іншими датчиками, такими як магнітometri і мікроконтролери [17].



### РОЗДІЛ 3. ПОБУДОВА КАНАЛУ ЗВ'ЯЗКУ ДЛЯ БОКСЕРСЬКОЇ ГРУШИ

Тренажер для боксу та інших видів єдиноборств відноситься до спорту і стосується конструкції навчально–тренувальних снарядів для контролю і відпрацювання ударів в боксі та інших видах силових єдиноборств. Використання бездротової технології підвищує мобільність спортсмена при тренуванні та спрощує реалізацію системи збору фізичних даних [21].

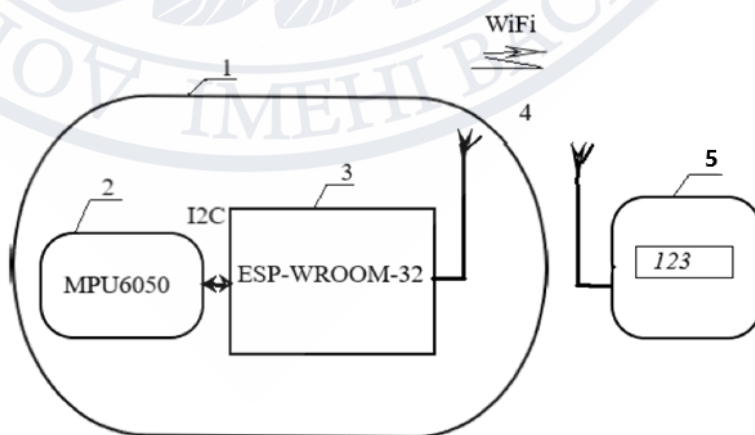
#### 3.1 Опис боксерської груші

На рисунку 3.1 представлена структурно–функціональна схема тренажеру, яка об'єднана в систему, де ESP32 використовується як сервер для передачі даних з MPU6050 через однорангову мережу до клієнта [21].

Передача даних з мікроконтролера, відбувається всередині локальної мережі, до якої мікроконтролер має бути попередньо підключений [21].

Доступ до даних, користувачем, відбувається через веб-сторінку за адресою мікроконтролера в мережі [21].

Груша містить ударний снаряд з наповнювачем, усередині якого розташований керуючий мікроконтролер, наповнена тирсою. В порожнині розміщені датчик, який вимірює показники удару. При використанні дротового зв'язку з вимірювальною апаратурою, виникає неможливість оцінки результатів на відстані [21].



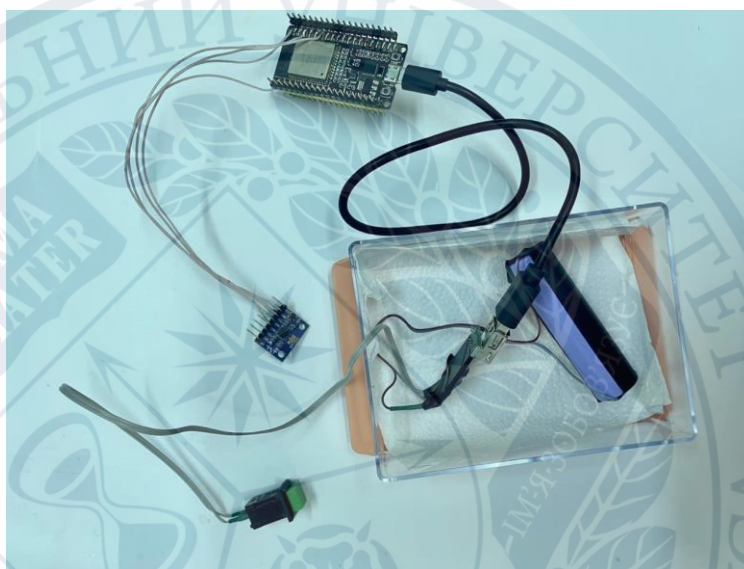
**Рисунок 3.1** – Структурно–функціональна схема для боксу

1 – Ударний снаряд (боксерська груша);



- 2 – Трьохосьовий гіроскоп з трьохосьовим акселерометром MPU6050;
- 3 – Мікроконтролер ESP-WROOM-32;
- 4 – Пристрій прийому і обробки бездротового сигналу Wi-Fi;
- 5– Комп'ютер.

Пристрій (рис. 3.1) складається зовнішнього чохла (рис. 3.3) з натуральної шкіри, в який поміщений пристроєве наповнення груші (рис. 3.2). Зовнішній чохол наповнений тирсою.



**Рисунок 3.2 – Пристроєве наповнення груші**

В середині груші розташовані датчик гіроскоп з акселерометром MPU6050 (2), елементи електроніки, допоміжні елементи, що скріплює конструкцію, а також батарейка під'єднана до ESP 32, який має вбудований модуль Wi-Fi. Плата бездротового зв'язку (4) передає інформацію, що знімається з датчика (2) на мікроконтролер (3), який передається по Wi-Fi на комп'ютер, що управляє (5) [21].



**Рисунок 3.3 – Макет боксерської груші**

### **3.2 Огляд проекту**

Спрощення пристрою при одночасному збереженні його функціональних можливостей досягається за рахунок того, що вимірювальна система тренажера складається з трьохосьового гіроскопа з трьохосьового акселерометра MPU6050 підключеним до мікроконтролеру ESP32-WROOM-32 [21].

MPU6050 відомий як 6-осьовий датчик або датчик 6-ти ступенів свободи (DOF). У цьому єдиному модулі присутні датчики акселерометра та гіроскопа. Датчик акселерометра видає вихідні показники з точки зору сили, прикладеної до об'єкта за рахунок сили тяжіння, а датчик гірометра видає результати щодо кутового зміщення об'єкта за годинниковою стрілкою або проти годинникової стрілки [19].

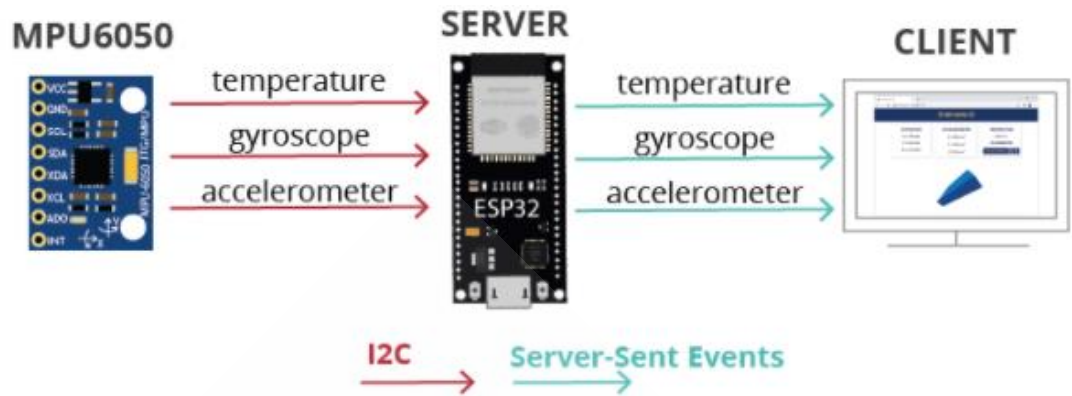
У цьому розділі ми побудуємо веб-сервер з ESP32 для відображення показань з акселерометра MPU-6050 та датчика гіроскопа. Ми також створимо 3D-зображення орієнтації датчика у веб-браузері. Показання оновлюються автоматично за допомогою переданих сервером подій, а тривимірне

представлення обробляється за допомогою бібліотеки JavaScript під назвою three.js [16]. Плата ESP32 буде запрограмована з використанням ядра Arduino [20].

Для побудови веб-сервера ми використовуватимемо бібліотеку ESPAsyncWebServer, яка забезпечує простий спосіб побудови асинхронного веб-сервера та обробки переданих сервером подій [16].

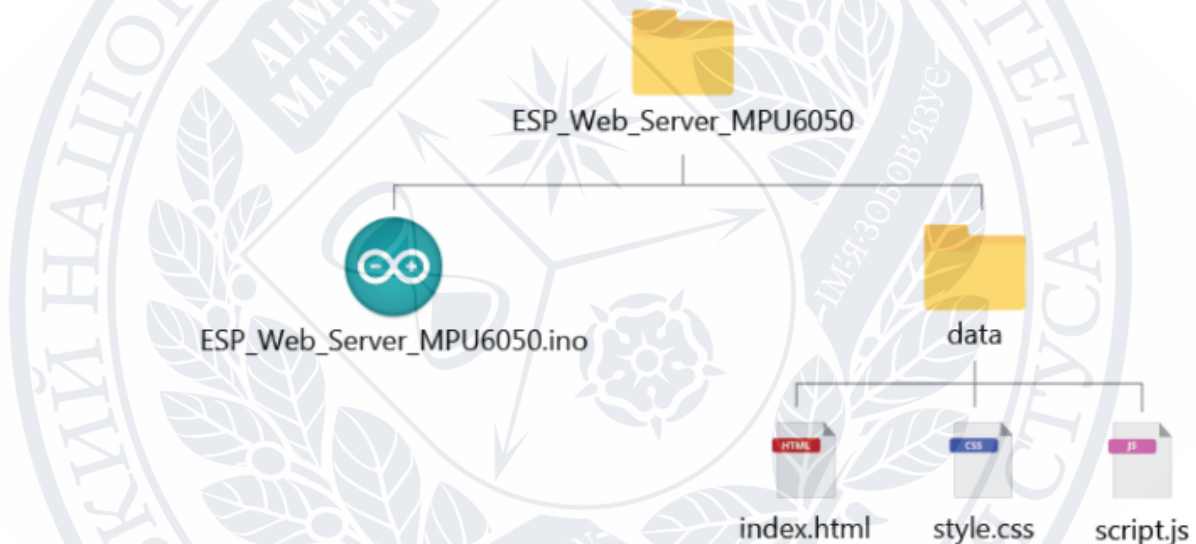
- Веб-сервер відображає значення гіроскопа осі X, Y і Z;
- Значення гіроскопа оновлюються на веб-сервері кожні 10 мілісекунд;
- Він відображає значення акселерометра (X, Y, Z). Ці значення оновлюються кожні 200 мілісекунд;
- Модуль датчика MPU-6050 також вимірює температуру, тому ми також відображатимемо значення температури. Температура оновлюється щосекунди (1000 мілісекунд);
- Усі показання оновлюються за допомогою Поділяються сервером подій;
- Існує 3D-зображення датчика Орієнтація 3D-об'єкта змінюється відповідно до орієнтації датчика. Поточне положення датчика обчислюється з використанням значень гіроскопа;
- Об'єкт 3D створюється за допомогою бібліотеки JavaScript під назвою three.js [16] ;
- Є чотири кнопки для регулювання положення 3D-об'єкта [16]:
  - RESET POSITION: встановлює кутове положення на 0 по всій осі;
  - X: встановлює кутове положення X на нуль;
  - Y: встановлює кутове положення Y на нуль;
  - Z: встановлює кутове положення Z на нуль [16];





**Рисунок 3.4** – Передача даних на веб-сторінку [16]

Щоб організувати наш проект і полегшити його розуміння, ми створимо чотири різні файли для побудови веб-сервера [16]:



**Рисунок 3.5** – Створення чотирьох різних файлів для побудови веб-сервера [16]

- Код Arduino, який обробляє веб-сервер;
- HTML-файл: для визначення вмісту веб-сторінки;
- CSS-файл: для оформлення веб-сторінки;
- Файл JavaScript: для програмування поведінки веб-сторінки (обробляти відповіді веб-сервера, події та створювати 3D-об'єкт) [16].

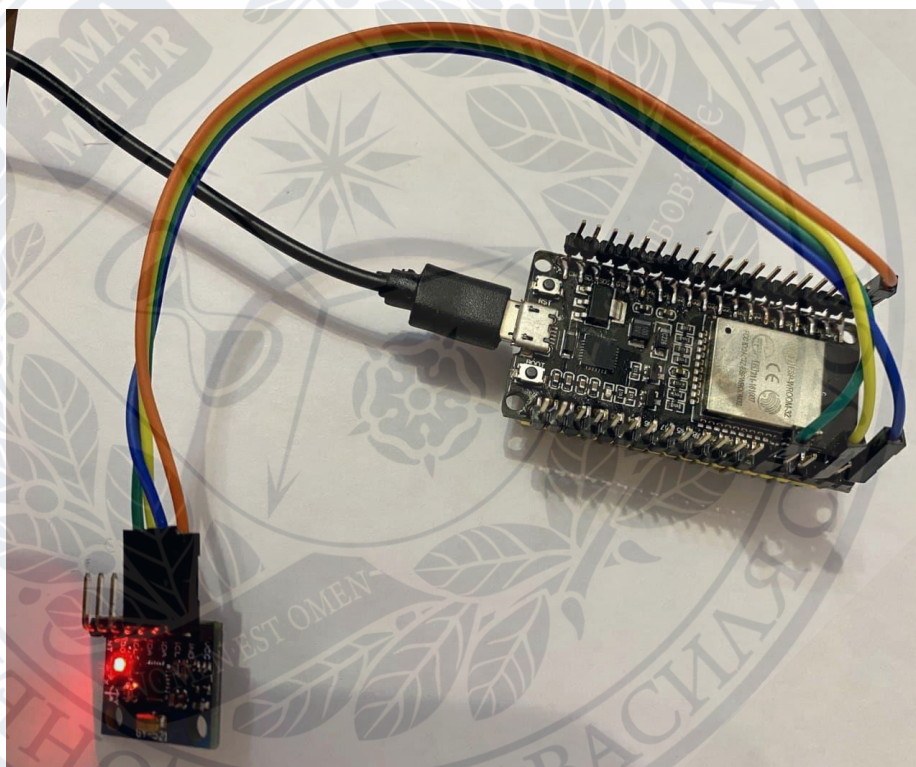
Файли HTML, CSS та JavaScript будуть завантажені до ESP32 SPIFFS (файлова система). Для завантаження файлів у файлову систему ESP32 ми будемо використовувати плагін завантажувача SPIFFS [16].

Для цього проекту потрібні такі компоненти:

1. Гіроскоп акселерометра MPU-6050 (керівництво ESP32);
2. ESP32;
3. Макет;
4. Провід перемички.

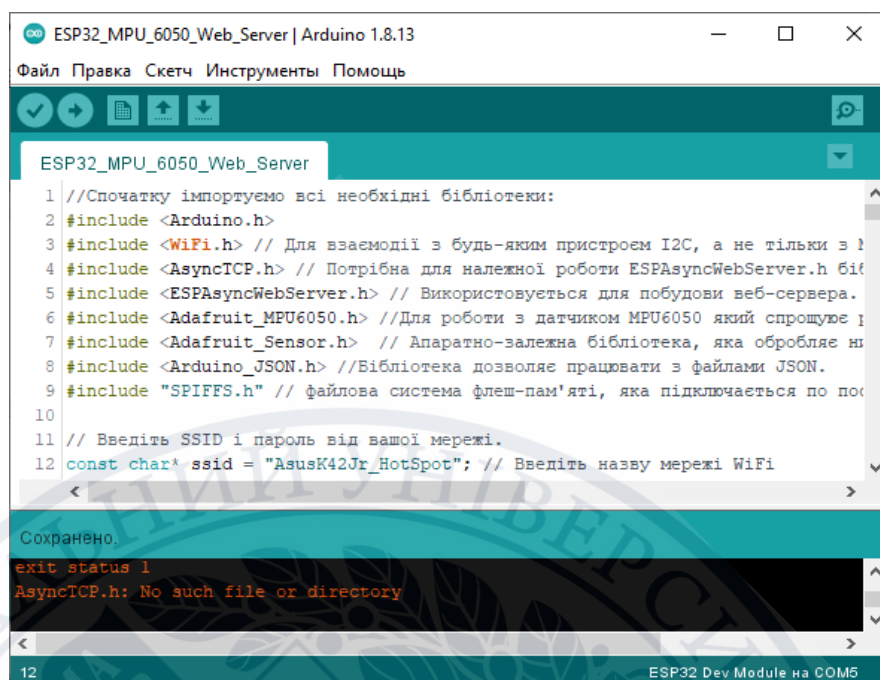
Принципова схема для взаємодії акселерометра та модуля гіроскопа MPU6050 з ESP-WROOM-32 дуже проста і була розглянута в 2 розділі.

Підключивши ESP-WROOM-32 до ноутбука і завантаживши код потрібно не забути правильно вибрати плату і порт. Переконайтеся, що ноутбук підключений до тієї ж мережі Wi-Fi, що і ESP-WROOM-32.



**Рисунок 3.6** – Підключення ESP-WROOM-32 до ноутбука

Вибрано програмне забезпечення Arduino з відкритим кодом (IDE). Він дозволяє легко писати код та завантажувати код на плату. З платою ESP-WROOM-32 будемо використовувати програмне забезпечення, яку будемо програмувати на C / C ++.





JavaScript слід зберігати всередині папки, що називається даними, всередині папки ескізів Arduino [16].

2. Після створення папки потрібно створити файл index.html із таким вмістом, який вказаний в Додатку Б.

3. Створений файл із назвою style.css із таким вмістом відповідає за стиль веб-сторінки (див. Додаток В).

4. Оскільки ми будемо використовувати зовнішній файл JavaScript з усіма функціями для обробки елементів HTML і створення 3D-анімації, нам потрібно створити файл script.js із вмістом, який показано в Додатку Г [16].

5. Щоб налаштувати сервер (ESP32), нам потрібно працювати в файлі - Ескіз Arduino (див. Додаток Д). Спочатку імпортуйте всі необхідні бібліотеки для цього проекту [16]:

```
#include <Arduino.h>
#include <Wi-Fi.h>
#include <AsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Arduino_JSON.h>
#include "SPIFFS.h"
```

6. Перш за все потрібно переконайтеся, чи вставлені наші мережеві дані для таких змінних. Вставити свої облікові дані мережі у такі змінні [16]:

```
// Replace with your network credentials

const char* ssid = "REPLACE_WITH_YOUR_SSID"; // Введіть назву
мережі Wi-Fi

const char* password = "REPLACE_WITH_YOUR_PASSWORD"; //
Введіть пароль мережі Wi-Fi
```

7. Створюємо AsyncWebServer об'єкт на порту 80 [16].

Створіть AsyncWebServer об'єкт на порту 80.

```
AsyncEventSource events("/events"); //створює нове джерело подій на
/events.
```

8. Оголошення змінних.

```
JSONVar readings;
// Timer variables
unsigned long lastTime = 0;
unsigned long lastTimeTemperature = 0;
unsigned long lastTimeAcc = 0;
unsigned long gyroDelay = 10;
unsigned long temperatureDelay = 1000;
unsigned long accelerometerDelay = 200;
```

9. Створено Adafruit\_MPU5060 виклик об'єкта mpu, створено події для показань датчика та змінних для збереження показань [16].

```
// Create a sensor object
Adafruit_MPU6050 mpu;
sensors_event_t a, g, temp;
float gyroX, gyroY, gyroZ;
float accX, accY, accZ;
float temperature;
```

10. Відрегулювання зміщення датчика гіроскопа по всій осі.

```
//Gyroscope sensor deviation
float gyroXerror = 0.07;
float gyroYerror = 0.03;
float gyroZerror = 0.01;
```

### 11. Ініціалізація MPU-6050

```
// initMPU () функція ініціалізує датчик MPU-6050
void initMPU(){
    if (!mpu.begin()) {
        Serial.println("Failed to find MPU6050 chip");
        while (1) {
            delay(10); } }
    Serial.println("MPU6050 Found!");}
```

### 12. Ініціалізація SPIFFS

```
void initSPIFFS() {
    if (!SPIFFS.begin()) {
        Serial.println("An error has occurred while mounting SPIFFS"); }
    Serial.println("SPIFFS mounted successfully");}
```

### 13. Ініціалізація Wi-Fi

```
// initWi-Fi () функція підключає ESP32 до вашої локальної мережі.
void initWi-Fi() {
    Wi-Fi.mode(WI-FI_STA);
    Wi-Fi.begin(ssid, password);
    Serial.print("Connecting to Wi-Fi ..");
    while (Wi-Fi.status() != WL_CONNECTED) {
        Serial.print('.');
        delay(1000); }
    Serial.println(Wi-Fi.localIP());}
```

14. Після отримання показників гіроскопа, акселерометра, температури, setup(), ініціалізується послідовний монітор, Wi-Fi, SPIFFS і датчик MPU [16].

```
void setup() {
```



```
Serial.begin(115200);

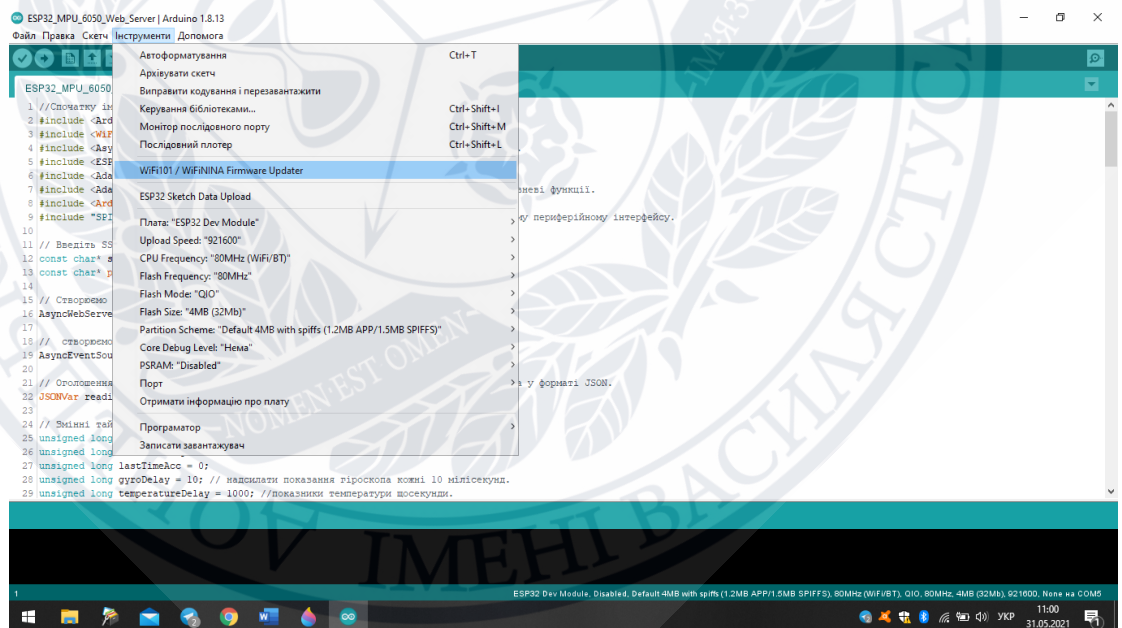
initWi-Fi();

initSPIFFS();

initMPU();
```

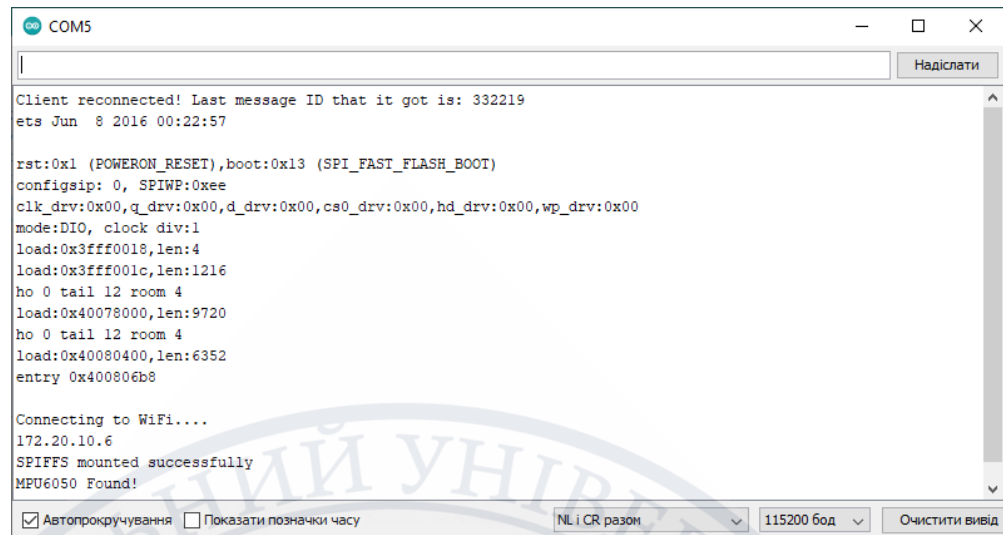
15. Оброблення запитів та надсилання подій клієнту з новими показаннями датчика [16].

16. Після вставки облікових даних мережі зберігаємо код. В програмі Arduino потрібно перейти до Ескізу та створити папку з назвою дані. Усередині цієї папки слід зберегти файли HTML, CSS та JavaScript. Далі завантажуюмо код на плату ESP32. Вибираємо правильну плату та COM-порт. Крім того, потрібно переконайтеся, що ми додали в код наші облікові дані мережі. Після завантаження коду потрібно завантажити файли. Переходимо до Інструменти > Завантаження ескізу даних ESP32 і дочекайтеся завантаження файлів [16].



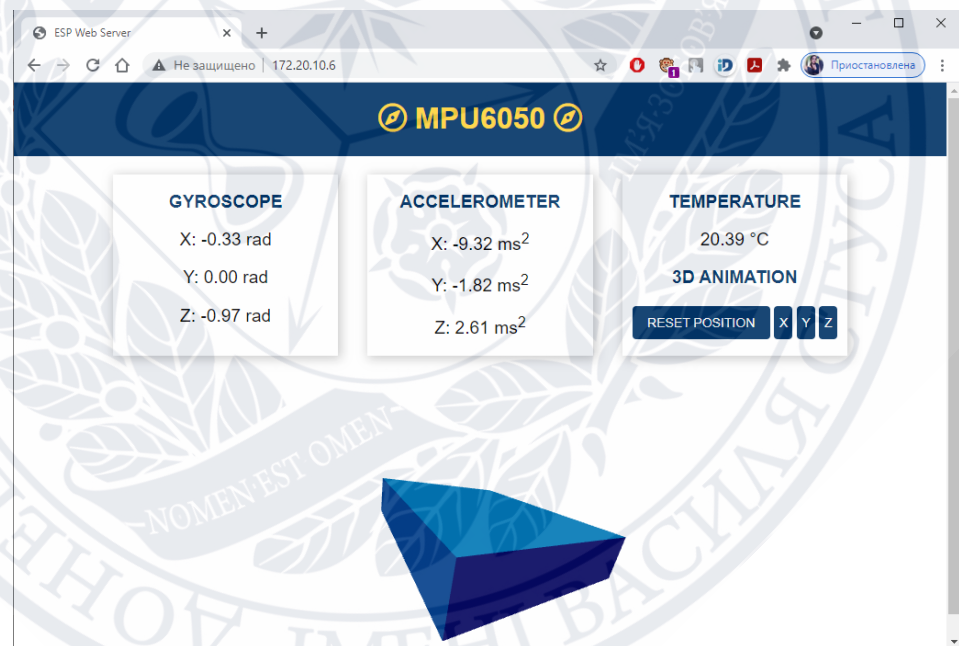
**Рисунок 3.8** – Перехід до ескізу та створення папки з назвою дані

17. Коли все буде успішно завантажено, відкриваємо послідовний монітор (COM 5) зі швидкістю передачі 115200. Натискаємо кнопку ESP32 EN / RST, і він повинен надрукувати IP-адресу ESP32. На монітор порта виводиться IP-адрес 172.20.10.6 з допомогою якого можна перейти на веб-сторінку [16].



**Рисунок 3.9** – Виведення IP-адресу на монітор порта

18. Демонстрація. Відкриваємо браузер і вводимо IP-адресу ESP32. Ми повинні отримати доступ до веб-сторінки, яка показує показання датчика [16].



**Рисунок 3.10** – Виведення даних на веб-сторінку

Датчик трохи дрейфує по осі X, незважаючи на деякі зміни в коді - це нормально для такого роду мікроконтролерів. Щоб зменшити дрейф, пропонується використовувати додатковий фільтр або фільтр Кальмана [16].

Лістинг програмної реалізації в повному обсязі представлений в Додатку Д.

### 3.2 Як працює тренажер для боксу

Спортсмен завдає удар (серію ударів) по боксерській груші (зовнішній чохол з натуральної шкіри). Ударний імпульс передається внутрішньому мішку, з огляду на, що останній заповнений тирсою. Отже, відсутні затухаючі коливання, що спотворюють характеристики ударного імпульсу (величину сили удару, час удару, вектор напрямку), ударний імпульс передається на датчик (2). Данні вимірювань з датчика надходять на мікроконтролер (3). Далі вони перетворюються для передачі по бездротовому каналу зв'язку на зовнішні пристрої. Бездротовий сигнал надходить на приймач сигналів ESP-WROOM-32 (3), який передається по Wi-Fi на комп'ютер, що управляє (7) [21].

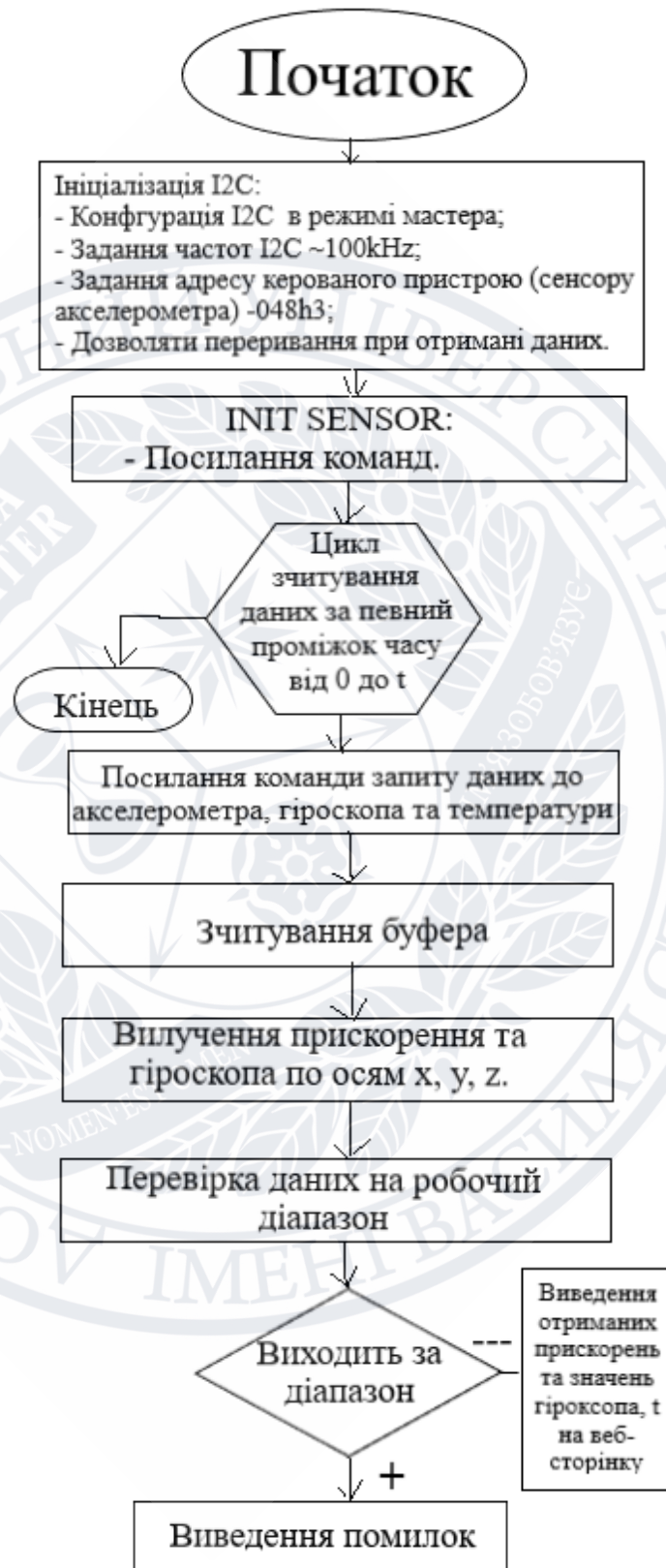
Прискорення мішка за трьома координатами вимірюються за допомогою тривісного інтегрального акселерометра і гіроскопа Д1. Датчик з необхідними метрологічними характеристиками і з відповідним розташуванням в просторі у внутрішній порожнині мішка досить для вимірювання значень цікавлять величин з необхідною точністю і розрахунку всіх похідних параметрів [18]. Як мікроконтролера в системі використаний цифровий сигнальний процесор ESP-WROOM-32 з низьким енергоспоживанням, з вбудованим 10-ти розрядним АЦП послідовних наближень (200 kps) і великим об'ємом пам'яті. Швидкодія процесора досить високо для обробки і попередньої фільтрації сигналів датчиків [21].

Обрана технологія Wi-Fi в якості бездротової технології передачі даних в складі системи. Застосування технології Wi-Fi є оптимально з точки зору розв'язуваної задачі в даній системі.

Таким чином, боксерська груша дозволяє підвищити ефективність процесу тренування за рахунок підвищення об'єктивності та точності вимірювання параметрів удару боксера і в той же час вона є більш простий і функціональної в порівнянні з прототипом. Виведення інформації в реальному масштабі часу як тренеру, так і спортсмену, дозволяє оптимізувати процес проведення тренування, а ведення докладної бази даних дає можливість мати науково—



обґрунтоване проектування тренувального і перед конкурентного періоду і підводити спортсмена до відповідальних змагань на піку форми [21].



**Рисунок 3.11** – Алгоритм отримання даних від модуля MPU6050 у мікроконтролер ESP-WROOM-32

## ВИСНОВКИ

1. Проаналізовані сучасні підходи бездротової передачі даних по безпроводним каналам зв'язку, та вибрано найбільш відповідну мережу Wi-Fi, тому що вона виявилася легше і з нею краще працювати.

2. Обрано за проведеним аналізом найбільш оптимальне рішення для збору та передачі даних з тренажера. Кожний з вибраних варіантів мав свої переваги та недоліки, але необхідно було вибрати кращий варіант не дивлячись на її недосконалість. Вирішальним питанням стали максимально зручне підключення, дешевизна та простота реалізації.

3. Побудовано канал бездротового зв'язку для передачі даних з датчиків за обраною технологією Wi-Fi у тренажері «Боксерська груша».

## СПИСОК ВИКОРИСТАНИХ ПОСИЛАНЬ

1. Никулин М., Мильченко В. Первая миля. Выпуск #2/2013. Универсальные системы – основа модернизации технологических сетей.
2. Коваленко И. Н. КОМПЬЮТЕРНЫЕ СЕТИ КОНСПЕКТЫ ЛЕКЦИЙ для студентов специальности 090205 Прикладная информатика (по отраслям) профессиональных образовательных учреждений Каменск–Шахтинский 2016.
3. Лисецкий Ю. М. Каналы связи как средство интеграции территориально распределенных структур. Информационные технологии и системы 2014.
4. Гейер, Джим. Беспроводные сети. Первый шаг : Пер. с англ. — М. : Издательский дом "Вильямс", 2005. — 192 с
5. Fantsiferov, Как выбрать стандарт связи для сети IoT, Москва Россия, 12 октября 2012 г. URL: <https://habr.com/ru/company/commandspot/blog/390825/>, (дата звернения: 01.06.2021).
6. Колыбельников А. И. ТРУДЫ МФТИ Обзор технологий беспроводных сетей Том 4, № 2 2012
7. Wireless Connectivity Guide. Texas Instruments, 2014.
8. Владимир Петриго, Беспроводные сети ZigBee. Часть 1 Санкт-Петербург Россия, 1 августа 1991 г. URL: <https://habr.com/ru/company/efo/blog/281048/>, (дата звернения: 01.06.2021).
9. What is a WiFi Technology & How Does It Work? 2013 – 2021. URL: <https://www.elprocus.com/how-does-wifi-technology-work/>, (дата звернения: 01.06.2021).
10. Steve Cope, Home Wireless Networking (Wi-Fi) for Beginners, February 9, 2021. URL: <https://stevessmarthomeguide.com/home-wireless-networking/>, (дата звернения: 01.06.2021).
11. Different Types of Wireless Communication Technologies January 5, 2021. URL: <https://www.watelectronics.com/different-types-wireless-communication-technologies/>, (дата звернения: 01.06.2021).



12. Insight Into ESP32 Features & Using It With Arduino IDE, 2021. URL: <https://lastminuteengineers.com/esp32-arduino-ide-tutorial/>, (дата звернення: 01.06.2021).
13. Electronic Components Datasheet Search. URL: [https://www.alldatasheet.com/view\\_datasheet.jsp?Searchword=ESP32](https://www.alldatasheet.com/view_datasheet.jsp?Searchword=ESP32), (дата звернення: 01.06.2021).
14. Tensilica, 23 March 2021. URL: <https://en.wikipedia.org/wiki/Tensilica>, (дата звернення: 01.06.2021).
15. MPU6050 (Gyroscope + Accelerometer + Temperature) Sensor Module, 2020. URL: <https://www.electronicwings.com/sensors-modules/mpu6050-gyroscope-accelerometer-temperature-sensor-module>, (дата звернення: 01.06.2021).
16. ESP32 Web Server with MPU-6050 Accelerometer and Gyroscope (3D object representation), 2013-2021. URL: <https://randomnerdtutorials.com/esp32-mpu-6050-web-server/>, (дата звернення: 01.06.2021).
17. Interfacing ESP32 with MPU6050, 2021. URL: [https://www.tutorialspoint.com/esp32\\_for\\_iot/interfacing\\_esp32\\_with\\_mpu6050.htm](https://www.tutorialspoint.com/esp32_for_iot/interfacing_esp32_with_mpu6050.htm), (дата звернення: 01.06.2021).
18. Динамометрический тренажер для боксу та інших видів єдиноборств: пат. 2292932 Россия: МКІ А63В 69/20, А63В 69/34, А63В 24/00 А.Н.Кудрін. – №2006104295; заяв. 14.02.2006; опубл. 10.02.2007 – прототип, Бюл. №12.
19. Rachna Aggarwal, Connecting MPU6050 With, 2021. URL: <https://www.instructables.com/Connecting-MPU6050-With-ESP32/>, (дата звернення: 01.06.2021).
20. Downloads Arduino IDE 1.8.15, 2021. URL: <https://www.arduino.cc/en/software>, (дата звернення: 01.06.2021).
21. ТРЕНАЖЕР ДЛЯ БОКСА И ДРУГИХ ВИДОВ ЕДИНОБОРСТВ: пат. 100729 Россия: МПК А63В 69/20/ Бакулев Сергей Евгеньевич.-№ 2010128809; заяв. 2010.07.12; опубл. 01.06.2021, Бюл. №12.

## ДОДАТОК А

## Далекий радіус дії та близький радіус дії

Таблиця А.1.1 – Далекий радіус дії

Дальній радіус дії												
Технічні характеристики	LoRaWAN	Стриж	SigFox	Wheatghless-N	Wheatghless-P	Wheatghless-W	Nuel	Nwave	Dash7	GSM, LTE	NB-CIoT	NB-LTE
Дальність	2–5 км в місті, до 45 км поза містом	До 40 км	10 км в місті, 50 км поза містом	5 км	2 км	5 км	10 км	10 км	До 2 км в місті	35 км GSM, 200 км LTE	–	–
Частота	Різна, менше 1 ГГц	868,8 МГц	Частотно незалежна мережа	Менше 1 ГГц	Менше 1 ГГц	Менше 1 ГГц	ISM або не використовувані пробіли частот		Різна, менше 1 ГГц	Ліцензійна, 8 – 900 МГц GSM, 7–900 МГц LTE	Ліцензійна, 7 – 900 МГц	Ліцензійна, 7 – 900 МГц

Продовження таблиці А.1.1

Швидкість передачі	0,3–50 кбіт/с, адаптивно	50 б/с	10–1000 біт/с	30–100 кбіт/с	До 100 кбіт/с, адаптивно	До 100 кбіт/с, адаптивно	10–100 кбіт/с		167 кбіт/с	70 кбіт/с GSM, до 1 Мбіт/с LTE	UL 1–32 кбіт/с, DL 1–20 кбіт/с	UL 1–144 кбіт/с, DL 1–200 кбіт/с
Споживання енергії	Низьке	Низьке	Низьке	Низьке	Низьке	Низьке	Низьке	Низьке	Низьке	Середньо низька	Низьке	Низьке
Аутентифікація	Присутня	Присутня	Присутня	Присутня	Присутня	Присутня	–	Присутня	Присутня	Високий рівень безпеки.	Високий рівень безпеки.	Високий рівень безпеки.
E2E шифрування	Присутнє	Присутнє	Присутнє	Присутнє	Присутнє	Присутнє	–	Присутнє	Присутнє	Присутнє	Присутнє	Присутнє
Діапазон ISM	Так	Так	Так	Так	Так	Так	Так, в залежності від базової станції	Так	Так	Ні		Ні



## Продовження таблиці А.1.1

Робота при значній забудові	Так	Так	Так	Так	Так	Так	ISM або не використовувані пробіли частот	Так	Так	Покращення		Так
Повна двунаправленість	Так, в залежності від режиму	Присутня	Ні	Ні	Присутня	Присутня	Присутня	Ні	Присутня	Присутня	Присутня	Присутня
Апгрейд ПО «повітрям»	Присутня	Ні	Ні	Ні	Присутня	Присутня	—	Ні	Присутня	Присутня	Присутня	Присутня
Підтримка сенсорів, рухомих між хабами	Присутня	Присутня	Ні	Так	Присутня	Присутня	—	Ні	Присутня	Присутня	Присутня	Присутня
Місцезнаходження сенсора відомо	Присутня	Тріангуляція по RSSI	Ні	Ні	Присутня	Присутня	—	Ні	Присутня	Присутня	Присутня	Присутня

Продовження таблиці А.1.1

Стандарт	LoRaWAN	Hi	Hi	Wheithless	Wheithless	Wheithless	Wheithless	Wheithless	ISO/IEC 18000-7	GSM Rel. 13; LTE-M Rel. 13	Можли во 3GPP Rel. 13 в березні 2016 р.	Можли во 3GPP Rel. 13 в березні 2016 р.
Масштабованість	Так	Так	Так	Обмежено	Так	Так	Так	Так	Так	Так	Так	Так

Таблиця А.1.2 – Близький радіус дії

Ближній радіус дії							
Технічні характеристики	Wireless RF	Bluetooth Low Energy	Wi-Fi	Wi-Fi HaLow	Thread	ZigBee	Z-Wave
Дальність	До 100 м в прямій відимості	80 м	До 100 м	До 1 км	Mesh	100 м / Mesh	30 м / Mesh
Частота	315/433MHz	2,4 ГГц	2,4 ГГц, 5 ГГц	900 ГГц	2,4 ГГц	915 МГц 2,4 ГГц	900 МГц
Швидкість передачі	10–115,2 кбіт/с	< 1 Мбіт/с	7 Гбіт/с максимально (Wi-Fi 802.11 ac)	50 кбіт/с – 18 Мбіт/с	250 кбіт/с	250 кбіт/с	10–100 кбіт/с

Продовження таблиці А.1.2

Споживання енергії	Наднизьке	Знижене	Високе	Знижене	Низьке	Низьке	Низьке
Аутентифікація	Так	Проблематично	Так	Так	Так	Так	Так
E2E шифрування	Так	Так	Так	Так	Так	Так	Так
Діапазон ISM	Частково	Так	Так	Так	Так	Так	Так
Робота при значній забудові	Ні	Ні	Ні	Так	Ні	Ні	Ні
Повна двунаправленість	–	Так	Так	Так	–	Так	Так
Апгрейд ПО «повітрям»	–	Так	Так	Так	–	Так	Так
Підтримка сенсорів, рухомих між хабами	–	Так	Так	Так	Ні	Так, mesh	Так, mesh
Місцезнаходження сенсора відомо	–	Ні	Так	Так	–	–	–
Стандарт	–	Bluetooth 4.0	IEEE 802.11	IEEE 802.11ah	На основі 6LoWPAN IEEE 802.15.4, Thread	На основі IEEE 802.15.4, ZigBee	Z-Wave
Масштабованість	Обмежено	Так	Так	Так	Так	Так	Обмежено



## ДОДАТОК Б

### Програма HTML

```

<!DOCTYPE HTML><html>
<head>
  <title>ESP Web Server</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" href="data:,">
  <link rel="stylesheet" type="text/css" href="style.css">
  <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.2/css/all.css" integrity="sha384-fnmOCqbTlWIlj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr" crossorigin="anonymous">
  <script src="https://cdnjs.cloudflare.com/ajax/libs/three.js/107/three.min.js"></script>
</head>
<body>
  <div class="topnav">
    <h1><i class="far fa-compass"></i> MPU6050 <i class="far fa-compass"></i></h1>
  </div>
  <div class="content">
    <div class="cards">
      <div class="card">
        <p class="card-title">GYROSCOPE</p>
        <p><span class="reading">X: <span id="gyroX"></span> rad</span></p>
        <p><span class="reading">Y: <span id="gyroY"></span> rad</span></p>
        <p><span class="reading">Z: <span id="gyroZ"></span> rad</span></p>
      </div>
      <div class="card">
        <p class="card-title">ACCELEROMETER</p>

```

```

    <p><span          class="reading">X:          <span          id="accX"></span>
ms<sup>2</sup></span></p>
    <p><span          class="reading">Y:          <span          id="accY"></span>
ms<sup>2</sup></span></p>
    <p><span          class="reading">Z:          <span          id="accZ"></span>
ms<sup>2</sup></span></p>
</div>
<div class="card">
    <p class="card-title">TEMPERATURE</p>
    <p><span class="reading"><span id="temp"></span> &deg;C</span></p>
    <p class="card-title">3D ANIMATION</p>
    <button id="reset" onclick="resetPosition(this)">RESET POSITION</button>
    <button id="resetX" onclick="resetPosition(this)">X</button>
    <button id="resetY" onclick="resetPosition(this)">Y</button>
    <button id="resetZ" onclick="resetPosition(this)">Z</button>
</div>
</div>
<div class="cube-content">
    <div id="3Dcube"></div>
</div>
</div>
<script src="script.js"></script>
</body>
</html>

```

## ДОДАТОК В

### Програма для стилю веб-сторінки

```
html {  
  font-family: Arial;  
  display: inline-block;  
  text-align: center;  
}  
p {  
  font-size: 1.2rem;}  
body {  
  margin: 0;}  
.topnav {  
  overflow: hidden;  
  background-color: #003366;  
  color: #FFD43B;  
  font-size: 1rem;  
}  
.content {  
  padding: 20px;}  
.card {  
  background-color: white;  
  box-shadow: 2px 2px 12px 1px rgba(140,140,140,.5);}  
.card-title {  
  color:#003366;  
  font-weight: bold;}  
.cards {  
  max-width: 800px;  
  margin: 0 auto;  
  display: grid; grid-gap: 2rem;
```

```

grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));}

.reading {
  font-size: 1.2rem;}

.cube-content{
  width: 100%;
  background-color: white;
  height: 300px; margin: auto;
  padding-top:2%;}

#reset{
  border: none;
  color: #FEFCFB;
  background-color: #003366;
  padding: 10px;
  text-align: center;
  display: inline-block;
  font-size: 14px; width: 150px;
  border-radius: 4px;}

#resetX, #resetY, #resetZ{
  border: none;
  color: #FEFCFB;
  background-color: #003366;
  padding-top: 10px;
  padding-bottom: 10px;
  text-align: center;
  display: inline-block;
  font-size: 14px;
  width: 20px;
  border-radius: 4px;
}

```



## ДОДАТОК Г

### Програма для обробки елементів HTML і створення 3D-анімації

```

let scene, camera, rendered, cube;

function parentWidth(elem) {
    return elem.parentElement.clientWidth;
}

function parentHeight(elem) {
    return elem.parentElement.clientHeight;
}

function init3D(){
    scene = new THREE.Scene();
    scene.background = new THREE.Color(0xffffff);
    camera = new THREE.PerspectiveCamera(75,
parentWidth(document.getElementById("3Dcube")) /
parentHeight(document.getElementById("3Dcube")), 0.1, 1000);
    renderer = new THREE.WebGLRenderer({ antialias: true });
    renderer.setSize(parentWidth(document.getElementById("3Dcube")),
parentHeight(document.getElementById("3Dcube")));
    document.getElementById('3Dcube').appendChild(renderer.domElement);
    // Create a geometry
    const geometry = new THREE.BoxGeometry(5, 1, 4);
    // Materials of each face
    var cubeMaterials = [
        new THREE.MeshBasicMaterial({color:0x03045e}),
        new THREE.MeshBasicMaterial({color:0x023e8a}),
        new THREE.MeshBasicMaterial({color:0x0077b6}),
        new THREE.MeshBasicMaterial({color:0x03045e}),
        new THREE.MeshBasicMaterial({color:0x023e8a}),
        new THREE.MeshBasicMaterial({color:0x0077b6}),
    ];

```

```

const material = new THREE.MeshFaceMaterial(cubeMaterials);
cube = new THREE.Mesh(geometry, material);
scene.add(cube);
camera.position.z = 5;
renderer.render(scene, camera);
}
// Resize the 3D object when the browser window changes size
function onWindowResize(){
    camera.aspect    =    parentWidth(document.getElementById("3Dcube"))    /
parentHeight(document.getElementById("3Dcube"));
    //camera.aspect = window.innerWidth / window.innerHeight;
    camera.updateProjectionMatrix();
    //renderer.setSize(window.innerWidth, window.innerHeight);
    renderer.setSize(parentWidth(document.getElementById("3Dcube")),
parentHeight(document.getElementById("3Dcube")));
}
window.addEventListener('resize', onWindowResize, false);
// Create the 3D representation
init3D();
// Create events for the sensor readings
if (!!window.EventSource) {
    var source = new EventSource('/events');
    source.addEventListener('open', function(e) {
        console.log("Events Connected");
    }, false);
    source.addEventListener('error', function(e) {
        if (e.target.readyState !== EventSource.OPEN) {
            console.log("Events Disconnected");
        }
    }, false);
}

```

```

source.addEventListener('gyro_readings', function(e) {
    //console.log("gyro_readings", e.data);
    var obj = JSON.parse(e.data);
    document.getElementById("gyroX").innerHTML = obj.gyroX;
    document.getElementById("gyroY").innerHTML = obj.gyroY;
    document.getElementById("gyroZ").innerHTML = obj.gyroZ;
    // Change cube rotation after receiving the readings
    cube.rotation.x = obj.gyroY;
    cube.rotation.z = obj.gyroX;
    cube.rotation.y = obj.gyroZ;
    renderer.render(scene, camera);
}, false);
source.addEventListener('temperature_reading', function(e) {
    console.log("temperature_reading", e.data);
    document.getElementById("temp").innerHTML = e.data;
}, false);
source.addEventListener('accelerometer_readings', function(e) {
    console.log("accelerometer_readings", e.data);
    var obj = JSON.parse(e.data);
    document.getElementById("accX").innerHTML = obj.accX;
    document.getElementById("accY").innerHTML = obj.accY;
    document.getElementById("accZ").innerHTML = obj.accZ;
}, false);
}

function resetPosition(element){
    var xhr = new XMLHttpRequest();
    xhr.open("GET", "/" + element.id, true);
    console.log(element.id);
    xhr.send();
}

```

## ДОДАТОК Д

### Програма для взаємодії ESP32 з MPU6050

//Спочатку імпортуємо всі необхідні бібліотеки:

```
#include <Arduino.h>
```

```
#include <Wi-Fi.h> // Для взаємодії з будь-яким пристроєм I2C, а не тільки з MPU6050;
```

```
#include <AsyncTCP.h> // Потрібна для належної роботи ESPAsyncWebServer.h бібліотеки.
```

```
#include <ESPAsyncWebServer.h> // Використовується для побудови веб-сервера.
```

```
#include <Adafruit_MPU6050.h> //Для роботи з датчиком MPU6050 який спрощує роботу.
```

```
#include <Adafruit_Sensor.h> // Апаратно-залежна бібліотека, яка обробляє низькоуровневі функції.
```

```
#include <Arduino_JSON.h> //Бібліотека дозволяє працювати з файлами JSON.
```

```
#include "SPIFFS.h" // файлова система флеш-пам'яті, яка підключається по послідовному периферійному інтерфейсу.
```

```
// Інформація про мережу Wi-Fi.
```

```
const char* ssid = "AsusK42Jr_HotSpot"; // Введіть назву мережі Wi-Fi
```

```
const char* password = "+380502601081"; // Введіть пароль мережі Wi-Fi
```

```
// Створюємо AsyncWebServer об'єкт на порту 80.
```

```
AsyncWebServer server(80);
```

```
// створюємо нове джерело подій на /events
```

```
AsyncEventSource events("/events");
```

```
// Оголошення змінних. Json Variable - це змінна JSON для зберігання показань датчика у форматі JSON.
```

```
JSONVar readings;
```

```
// Змінні таймера
```

```
unsigned long lastTime = 0;
```

```
unsigned long lastTimeTemperature = 0;
```



```

unsigned long lastTimeAcc = 0;
unsigned long gyroDelay = 10; // надсилати показання гіроскопа кожні 10
мілісекунд.
unsigned long temperatureDelay = 1000; //показники температури щосекунди.
unsigned long accelerometerDelay = 200; //показання акселерометра кожні 200
мілісекунд
// Створіть сенсорний об'єкт для збереження показань.
Adafruit_MPU6050 mpu;
sensors_event_t a, g, temp;
float gyroX, gyroY, gyroZ;
float accX, accY, accZ;
float temperature;
//Відрегулюйте зміщення датчика гіроскопа по всій осі.
float gyroXerror = 0.07;
float gyroYerror = 0.03;
float gyroZerror = 0.01;
// Ініціалізація MPU6050
void initMPU(){
  if (!mpu.begin()) {
    Serial.println("Failed to find MPU6050 chip");
    while (1) {
      delay(10);
    }
  }
  Serial.println("MPU6050 Found!");
}
// Ініціалізація SPIFFS
void initSPIFFS() {
  if (!SPIFFS.begin()) {
    Serial.println("An error has occurred while mounting SPIFFS"); }

```

```

Serial.println("SPIFFS mounted successfully");
}
// Ініціалізація Wi-Fi
void initWi-Fi() {
    Wi-Fi.mode(WI-FI_STA);
    Wi-Fi.begin(ssid, password);
    Serial.println("");
    Serial.print("Connecting to Wi-Fi...");
    while (Wi-Fi.status() != WL_CONNECTED) {
        Serial.print(".");
        delay(1000);
    }
    Serial.println("");
    Serial.println(Wi-Fi.localIP());
}
// getGyroReadings () функція отримує нові показники гіроскопа і повертає
поточну кутову орієнтацію на осі X, Y та Z як рядок JSON.
String getGyroReadings(){
    mpu.getEvent(&a, &g, &temp);
    // gyroX_temp змінна тимчасово містить поточне значення гіроскопа X.
    float gyroX_temp = g.gyro.x;
    // перевіряємо, чи значення датчика не перевищують зміщення.
    if(abs(gyroX_temp) > gyroXerror) {
        //застосування формули, щоб отримати кутове положення поточного датчика
        gyroX.
        gyroX += gyroX_temp/50.00; }
    // дотримуємося подібної процедури, щоб отримати значення Y та Z.
    float gyroY_temp = g.gyro.y;
    if(abs(gyroY_temp) > gyroYerror) {
        gyroY += gyroY_temp/70.00; }

```

```

float gyroZ_temp = g.gyro.z;
if(abs(gyroZ_temp) > gyroZerror) {
    gyroZ += gyroZ_temp/90.00; }
// об'єднуємо показання у змінну JSON (readings) і повернути рядок JSON
(jsonString).

readings["gyroX"] = String(gyroX);
readings["gyroY"] = String(gyroY);
readings["gyroZ"] = String(gyroZ);
String jsonString = JSON.stringify(readings);
return jsonString;}

//getAccReadings () повертає показання акселерометра.
String getAccReadings() {
    mpu.getEvent(&a, &g, &temp);
    // Get current acceleration values
    accX = a.acceleration.x;
    accY = a.acceleration.y;
    accZ = a.acceleration.z;
    readings["accX"] = String(accX);
    readings["accY"] = String(accY);
    readings["accZ"] = String(accZ);
    String accString = JSON.stringify (readings);
    return accString;}

// getTemperature () повертає поточне значення температури.
String getTemperature(){
    mpu.getEvent(&a, &g, &temp);
    temperature = temp.temperature;
    return String(temperature);}

// В setup(), ініціалізуйте послідовний монітор, Wi-Fi, SPIFFS і датчик MPU.
void setup() {
    Serial.begin(115200);

```

```

initWi-Fi();
initSPIFFS();
initMPU();
// Обробляти веб-сервер
// У своєму HTML-файлі ви посилаєтесь на файли style.css та script.js .
// Отже, коли файл HTML завантажується у ваш браузер, він зробить запит на
ці файли CSS та JavaScript.
server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send(SPIFFS, "/index.html", "text/html");
});
server.serveStatic("/", SPIFFS, "/");
// Коли натискаєте кнопку RESET POSITION, ESP32 отримує запит на/
скидання шляху.
server.on("/reset", HTTP_GET, [](AsyncWebServerRequest *request){
    // встановлення gyroX, gyroY, gyroZ до нуля, щоб відновити початкове
положення датчика.
    gyroX=0;
    gyroY=0;
    gyroZ=0;
    request->send(200, "text/plain", "OK");
});
server.on("/resetX", HTTP_GET, [](AsyncWebServerRequest *request){
    gyroX=0;
    request->send(200, "text/plain", "OK");
});
server.on("/resetY", HTTP_GET, [](AsyncWebServerRequest *request){
    gyroY=0;
    request->send(200, "text/plain", "OK");
});
server.on("/resetZ", HTTP_GET, [](AsyncWebServerRequest *request){

```



```

gyroZ=0;
request->send(200, "text/plain", "OK");
});
// Обробляємо події веб-сервера
events.onConnect([](AsyncEventSourceClient *client){
    if(client->lastId()){
        Serial.printf("Client reconnected! Last message ID that it got is: %u\n", client-
>lastId());  }
        // send event with message "hello!", id current millis
        // and set reconnect delay to 1 second
        client->send("hello!", NULL, millis(), 10000);
    });
server.addHandler(&events);
// Запуск сервера
server.begin();}
// В loop (), ми будемо надсилати події клієнту з новими показаннями датчика.
void loop() {
    if ((millis() - lastTime) > gyroDelay) {
        // Надсилайте події гіроскопа на веб-сервер із показаннями датчиків
        events.send(getGyroReadings().c_str(),"gyro_readings",millis());
        lastTime = millis(); }
    if ((millis() - lastTimeAcc) > accelerometerDelay) {
        // Надсилайте події акселерометра на веб-сервер із показаннями датчиків
        events.send(getAccReadings().c_str(),"accelerometer_readings",millis());
        lastTimeAcc = millis(); }
    if ((millis() - lastTimeTemperature) > temperatureDelay) {
        // Надсилайте події температури на веб-сервер із показаннями датчиків
        events.send(getTemperature().c_str(),"temperature_reading",millis());
        lastTimeTemperature = millis(); }
}

```