Міністерство освіти і науки України
Український державний університет науки і технологій

Факультет «Комп'ютерні технології і системи»
(назва факультету)

Кафедра «Електронні обчислювальні машини»
(повна назва кафедри)

Пояснювальна записка

до кваліфікаційної роботи
магістра
(ступінь вищої освіти)

на тему: Дослідження та розробка апаратно-програмних комплексів засобів генерації випадкових чисел. Комплекс генерації випадкових чисел на базі мобільних пристроїв

за освітньою програмою Комп'ютерна інженерія
зі спеціальності: 123 Комп'ютерна інженерія
(шифр і назва спеціальності)

Виконав: студент групи: КС2321

_____ / Артур ОПРЯТНИЙ /
(підпис студента) (Ім'я ПРІЗВИЩЕ)

Керівник: _____ / доцент, Денис ОСТАПЕЦЬ /
(підпис) (посада, Ім'я ПРІЗВИЩЕ)

Нормоконтролер: _____ / доцент, Олег ЄГОРОВ /
(підпис) (посада, Ім'я ПРІЗВИЩЕ)

Консультанти:

_____ _____ / /
(назва розділу) (підпис) (посада, Ім'я ПРІЗВИЩЕ)

_____ _____ / /
(назва розділу) (підпис) (посада, Ім'я ПРІЗВИЩЕ)

_____ _____ / /
(назва розділу) (підпис) (посада, Ім'я ПРІЗВИЩЕ)

Засвідчую, що у цій роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Дніпро – 2025 рік

Ministry of Education and Science of Ukraine
Ukrainian State University of Science and Technologies

Faculty «Computer technologies and systems»
(faculty)

Department «Electronic computers»
(department)

Explanatory Note
to Master's Thesis
first (master's)
(higher education degree)

on the topic: Research and development of hardware and software complexes of means for random numbers generating. Random number generating complex based on mobile devices

according to educational curriculum Computer Engineering

in the Speciality: 123 Computer Engineering
(speciality and its code)

Done by the student of the group: KC2321 / Artur Opriatnyi /
(name, surname)

Scientific Supervisor: / Associate Professor, Denys Ostapets /
(position, name, surname)

Normative controller : / Associate Professor, Oleg Yehorov /
(position, name, surname)

Supervisors

(Chapter title heading)	/ / (position, name, surname)
(Chapter title heading)	/ / (position, name, surname)
(Chapter title heading)	/ / (position, name, surname)
(Chapter title heading)	/ / (position, name, surname)

Dnipro – 2025

Міністерство освіти і науки України
Український державний університет науки і технологій

Факультет: Комп'ютерні технології і системи
Кафедра: ЕОМ
Рівень вищої освіти: Другий (магістерський)
Освітня програма: Комп'ютерна інженерія
Спеціальність: 123 Комп'ютерна інженерія
(шифр та назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри ЕОМ
_____ проф. Жуковицький
(підпис) (Ім'я ПРІЗВИЩЕ)
Дата _____

З А В Д А Н Н Я

на кваліфікаційну роботу магістра
(ступінь вищої освіти)

студенту Опрятному Артуру Олександровичу
(Прізвище, Ім'я По батькові)

1. Тема роботи: Дослідження та розробка апаратно-програмних комплексів засобів генерації випадкових чисел. Комплекс генерації випадкових чисел на базі мобільних пристроїв

Керівник роботи: Остапець Денис Олександрович, к.т.н., доцент
(Прізвище, Ім'я, По батькові, науковий ступінь, вчене звання)

затверджені наказом від "29" _03_ 2022 р. № _284_

2. Строк подання студентом роботи: 20.01.2025 р.
3. Вихідні дані до роботи: _____

4. Зміст пояснювальної записки (перелік питань, які потрібно опрацювати):
4.1 Огляд та аналіз способів генерації випадкових чисел на базі мобільних пристроїв;
4.2 Архітектура, функції та режими роботи комплексу;
4.3 Розробка програмного забезпечення комплексу;
4.4 Дослідження якості випадкових чисел, згенерованих за допомогою комплексу.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Завдання видав: (підпис консультанта, дата)	Завдання прийняв: (підпис студента, дата)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Огляд та аналіз способів генерації випадкових чисел на базі мобільних пристроїв		
2	Архітектура, функції та режими роботи комплексу		
3	Розробка програмного забезпечення комплексу		
4	Дослідження якості випадкових чисел, згенерованих за допомогою комплексу		
5	Реферат, вступ, висновки		5%
6	Подання кваліфікаційної роботи до кафедри	20.01.24	
7	Захист кваліфікаційної роботи на засіданні Екзаменаційної комісії	24.01.24	

Студент _____ (підпис) Артур ОПРЯТНИЙ (Ім'я ПРІЗВИЩЕ)

Керівник роботи _____ (підпис) Денис ОСТАПЕЦЬ (Ім'я ПРІЗВИЩЕ)

Міністерство освіти і науки України
Український державний університет науки і технологій

Відгук керівника
кваліфікаційної роботи магістра

Студент групи КС2321 Опрятний Артур Олександрович
(шифр групи) (Прізвище, Ім'я, По батькові)

Тема випускної роботи: Дослідження та розробка апаратно-програмних комплексів засобів генерації випадкових чисел. Комплекс генерації випадкових чисел на базі мобільних пристроїв

1. Якісні відмінності кваліфікаційної роботи:
В роботі виконано огляд та аналіз джерел ентропії, що доступні в мобільних пристроях для генерації випадкових чисел. У якості джерел ентропії обрано датчики акселерометра, гіроскопа та магнітометра. Розроблено архітектуру, функції та режими роботи комплексу. Описано принципи обміну даними між елементами комплексу та принципи генерації чисел. Розроблено програмне забезпечення серверної, клієнтської та мобільної частини комплексу. Виконане дослідження якості отримуваних чисел за допомогою статистичних та візуальних тестів. Проведено аналіз результатів.

Основні положення роботи доповідались та були схвалені на XV, XVI та XVIII Міжнародних науково-практичних конференціях «Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті» у 2021, 2022 та 2024 роках, а також на Всеукраїнській науково-технічній конференції студентів і молодих учених «Наука і сталий розвиток транспорту» у 2024 році. Опубліковано відповідні тези доповідей.

2. Зауваження:
Суттєві зауваження по роботі відсутні

3. Висновок щодо дотримання академічної доброчесності
Академічну доброчесність дотримано.

Комплексна оцінка кваліфікаційної роботи:
Дипломна робота заслуговує відмінної оцінки, а її автору може бути надано рекомендацію до вступу в аспірантуру.

Керівник: доцент каф. ЕОМ Денис ОСТАПЕЦЬ
Дата: 23.01.2025р.

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи магістра: 115 с., 55 рис., 8 табл., 13 додатків, 29 джерел.

Об’єкт розробки – апаратно-програмний комплекс генерації випадкових чисел на базі мобільного пристрою.

Мета роботи – розробка та дослідження ефективності апаратно-програмного комплексу генерації випадкових чисел на базі мобільного пристрою.

Методи дослідження – експериментальне дослідження якості отримуваних випадкових чисел з використанням кейсів статистичних та візуальних тестів.

Здійснено огляд на аналіз джерел ентропії, що доступні в мобільних пристроях для генерації випадкових чисел. У якості джерел ентропії обрано датчики акселерометра, гіроскопа та магнітометра. Розроблено архітектуру, функції та режими роботи комплексу. Описано принципи обміну даними між елементами комплексу та принципи генерації чисел. Розроблено програмне забезпечення серверної, клієнтської та мобільної частини комплексу. Досліджено якість отримуваних чисел за допомогою статистичних та візуальних тестів.

Розроблений комплекс може використовуватися на практиці для отримання випадкових та псевдовипадкових чисел та у навчальних цілях.

Ключові слова: ВИПАДКОВІ ЧИСЛА, ПСЕВДОВИПАДКОВІ ЧИСЛА, ДЖЕРЕЛО ЕНТРОПІЇ, АКСЕЛЕРОМЕТР, ГІРОСКОП, МАГНІТОМЕТР, TCP, GO, DART, FLUTTER, СЕРВЕР, КЛІЄНТ.

ЗМІСТ

ВСТУП

У сучасному світі генерація випадкових чисел є важливою складовою багатьох важливих процесів, зокрема в криптографії, моделюванні та іграх. Від якості генерації випадкових чисел залежить ефективність роботи багатьох алгоритмів і систем, що використовуються в зазначених сферах.

Для багатьох існуючих алгоритмів генерації псевдовипадкових чисел значним викликом є забезпечення їх високої якості та непередбачуваності. Відносно висока вартість та обмежена швидкодія апаратних засобів створення дійсно випадкових послідовностей чисел, є основними перешкодами на шляху до їх широкого застосування у критично важливих системах. Тому тема роботи є актуальною.

Мета роботи – розробка та дослідження ефективності апаратно-програмного комплексу генерації випадкових чисел на базі мобільного пристрою.

Основні положення роботи доповідались та були схвалені на XV, XVI та XVIII Міжнародних науково-практичних конференціях «Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті» у 2021, 2022 та 2024 роках, а також на Всеукраїнській науково-технічній конференції студентів і молодих учених «Наука і сталий розвиток транспорту» у 2024 році. Опубліковано відповідні тези доповідей [1, 2, 3, 4].

Представлена робота складається зі вступу, 4 розділів, висновків та додатків. У розділі 1 представлений огляд та порівняльний аналіз джерел ентропії що доступні у мобільних пристроях для генерації випадкових чисел. У розділі 2 розроблено узагальнену архітектуру комплексу та принципи взаємодіє всіх його елементів між собою. У розділі 3 представлено розробку програмного забезпечення серверної, клієнтської та мобільної частини комплексу. У розділі 4 наведено результати дослідження якості отримуваних випадкових чисел.

1 ОГЛЯД ТА АНАЛІЗ СПОСОБІВ ГЕНЕРАЦІЇ ВИПАДКОВИХ ЧИСЕЛ НА БАЗІ МОБІЛЬНИХ ПРИСТРОЇВ

1.1 Загальні відомості

Сучасні технології широко використовують випадкові послідовності чисел в різних галузях – від імітаційного моделювання до криптографії. При цьому дуже важливо використовувати якісні генератори випадкових чисел, так як від цього залежить якість одержуваних результатів [5]. За допомогою випадкових чисел можна реалізувати безліч завдань: тестування алгоритмів, моделювання, імітація введення користувача, деякі завдання чисельного аналізу та інші [6].

Джерело ентропії – це фізичне джерело інформації, вихід якого або здається випадковим, або стає таким після застосування певного процесу фільтрації/дистиляції [7].

Під мобільними пристроями автор розуміє смартфони, планшети та інші подібні засоби під керуванням мобільних операційних систем.

Датчики – це пристрої, які використовуються в мобільних пристроях для виявлення різних аспектів навколишнього середовища [8]. Зараз існує достатньо багато датчиків, доступних в смартфонах, які вбудовані та допомагають функціонувати смартфону. Здебільшого ці датчики створені для проведення оцінки досвіду користувачів. Генератор випадкових чисел, що використовуючи датчики сучасного мобільного пристрою, здатний створювати високоякісні послідовності випадкових бітів [9]. Основні датчики у смартфоні які можна використати для генерації випадкових чисел приведено на рис. 1.1.

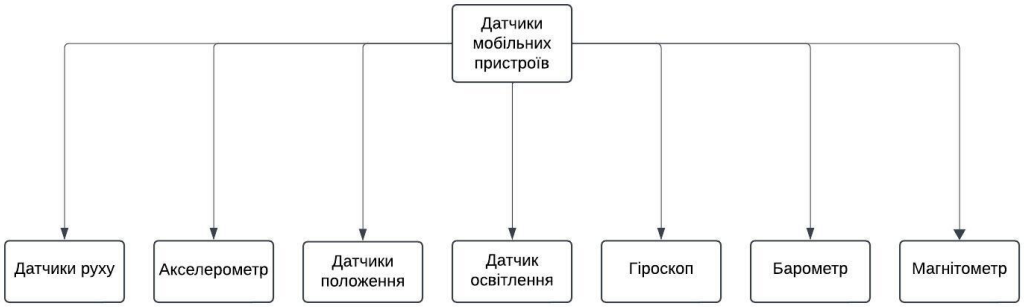

Рисунок 1.1 – Основні датчики, доступні у мобільних пристроях

Алгоритм генерації псевдовипадкових чисел – це процедура, що використовує математичні формули або заздалегідь розраховані значення для створення числових послідовностей, які можуть здатись випадковими на перший погляд. Найбільш поширеним прикладом такого методу є лінійний конгруентний алгоритм. Велика кількість наукових досліджень присвячена вивченню теорії псевдовипадкових чисел, а сучасні методи настільки досконалі, що згенеровані числа важко відрізнити від випадкових.

Відмінності між випадковими і псевдовипадковими числами легко зрозуміти, якщо порівняти згенеровані комп'ютером випадкові числа з кидками кубика. Оскільки генератори псевдовипадкових чисел генерують випадкові числа за допомогою математичних формул або попередньо обчислених списків, його використання полягає тому, що хтось багато разів кидає кубик і записує результати. Щоразу, коли ви просите кинути кубик, ви отримуєте наступне число у списку. По суті, числа здаються випадковими, але насправді вони заздалегідь визначені. Генератори випадкових чисел працюють змушуючи комп'ютер фактично кинути кубик – або, частіше, використовують інші фізичні явища, які легше підключити до комп'ютера, ніж кубик.

Генератори псевдовипадкових чисел є ефективними, так як вони можуть створювати велику кількість чисел за короткий час, і є детермінованими, тобто певну послідовність чисел можна відтворити пізніше, якщо відоме початкове число в послідовності. Генератори псевдовипадкових чисел зазвичай також є періодичними, що означає, що послідовність зрештою повториться. Хоча періодичність навряд чи є бажаною характеристикою, сучасні генератори мають такий довгий період, що його можна ігнорувати для більшості практичних цілей.

Ці характеристики роблять генератори псевдовипадкових чисел придатними для застосувань, де потрібно багато чисел і де корисно легко відтворювати ту саму послідовність. Популярними прикладами таких додатків є додатки для симуляції та моделювання. Але вони не підходять для програм, де важливо, щоб

числа були справді непередбачуваними, наприклад для шифрування даних та азартних ігор.

Фізичне явище може бути дуже простим, як-от невеликі варіації в рухах миші або в кількості часу між натисканнями клавіш. На практиці, однак, треба бути обережними з вибором джерела. Наприклад, може бути складно використовувати натискання клавіш таким чином, оскільки натискання клавіш часто буферизуються операційною системою комп'ютера, тобто збираються кілька натиснень клавіш, перш ніж вони надсилаються програмі, яка на них чекає. Програмі, яка очікує натискання клавіш, здасться, ніби клавіші були натиснуті майже одночасно, і врешті-решт тут може бути не так багато випадковості.

1.2 Джерела ентропії, які доступні у мобільних пристроях

1.2.1 Датчики руху

Датчики руху – датчики які використовуються для моніторингу руху пристрою. Дані рухи можуть бути такими: нахил, струс, обертання або коливання [10]. Смартфони ідентифікують свою орієнтацію за допомогою акселерометра. Датчики руху, присутні в акселерометрі, можуть використовуватися для виявлення землетрусів або в медичних пристроях.

У роботі [11], аналізуються дані, отримані з датчиків руху та інших різних типів датчиків, які знаходяться на більшості плат, що використовуються для вузлів IoT. Аналіз виконується за трьома сценаріями (нормальний, динамічний і насичення), залежно від специфіки кожного датчика. Щоб оцінити рівень ентропії, автори використали методологію NIST. Беручи до уваги те, що дані є упередженими, вони використали методологію оцінки, яка використовується для незалежно та однаково розподілених джерел ентропії з використанням чотирьох оцінок (частотний тест, тест на зіткнення, частковий збір і тест на стиснення), ентропію Шеннона та мінімальну ентропію. Значення вихідної ентропії вважалося мінімальним із значень ентропії, отриманих від оцінювачів. У результаті аналізу було виявлено, що ентропія Шеннона та мінімальна ентропія навіть у п'ять разів вищі, ніж отримані іншими оцінювачами. Це може пояснити

відмінності в значеннях ентропії, про які повідомляють інші автори. Хоча аналіз ентропії, проведений у цьому дослідженні, найбільш близький до методології NIST, кілька аспектів можуть впливати на правильність результатів. У цьому дослідженні було використано чотири оцінки порівняно з остаточною версією рекомендацій NIST, де використовуються десять оцінок. Кількість даних, зібраних для кожного експерименту, неоднакова, і це впливає на розрахункові значення ентропії. У деяких випадках з різних причин вони не можуть зібрати достатню кількість даних, необхідних для хорошої оцінки ентропії.

1.2.2 Акселерометр

Акселерометр – це прилад, що вимірює силу реакції, яка індукована прискоренням або гравітацією [10]. У мобільних пристроях акселерометри здебільшого використовуються для керування орієнтації зображення на екрані (книжкове чи альбомне). Ще одним способом застосування акселерометра може бути виконання мобільним пристроєм деяких функцій при зміні орієнтації у просторі (струшування, удар, поворот дисплею та деякі інші). Принцип використання датчик акселерометра зображений на рис. 1.2.

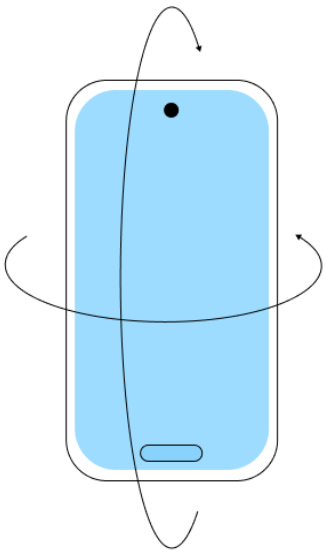

Рисунок 1.2 – Принцип використання акселерометра

У роботі [12] досліджується можливість реалізації генератора випадкових чисел з використанням даних, придбаних з акселерометра. Крім того, автори представляють рішення, реалізоване на RFID-мітці, пристрої з обмеженими ресурсами. Автори демонструють, що акселерометр генерує ентропію, навіть коли він використовується в стаціонарному режимі, і що він стійкий до різноманітних змін навколишнього середовища та агресивних маніпуляцій. Рівень ентропії розраховується для різних типів рухів, рівнів шуму, частот дискретизації, температур і навіть резонансної частоти датчика. Висновок полягає в тому, що найнижчий рівень ентропії досягається, коли датчик знаходиться в стаціонарному стані.

1.2.3 Датчики положення

Смартфон має два датчики, що дозволяють визначати фізичне положення пристрою – датчик магнітометра з комбінацією датчика акселерометра [13]. На рис. 1.3 зображено принцип використання датчика положення.

Рисунок 1.3 – Принцип використання датчика положення

Виробники смартфонів зазвичай використовують датчик положення, щоб з’ясувати, коли слухавка тримається близько до обличчя користувача, наприклад, під час розмови. Ці датчики корисні для визначення фізичного положення

пристрою. Наприклад, можна використовувати геомагнітний датчик в поєднанні з акселерометром для визначення положення пристрою відносно північного магнітного полюса [14].

1.2.4 Датчик освітлення

Датчик освітлення – це датчик, який регулює рівень яскравості екрана [15]. Він доступний як в дешевих, так і в дорогих смартфонах. Якщо ви перевели свій смартфон у режим автоматичної яскравості, тоді, коли ви виходите на світло, ваш телефон автоматично підвищуватиме яскравість екрана. Коли ви заходите в темряві, то за допомогою цього датчика яскравість телефону буде тьмяною. Залежно від інтенсивності світла, цей датчик керує яскравістю екрану. Принцип використання датчика освітлення приведено на рис. 1.4.

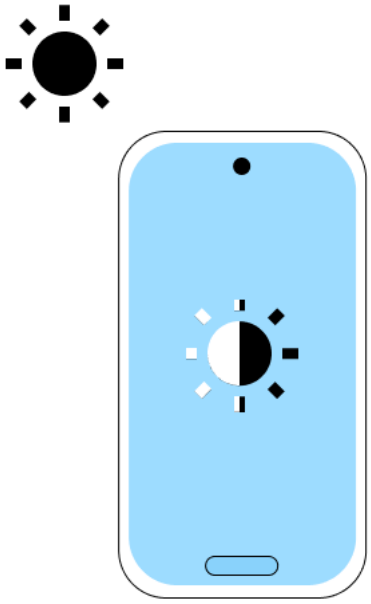

Рисунок 1.4 – Принцип використання датчика освітлення

У роботі [16], автори продемонстрували, що тип вимірювань освітлення навколишнього середовища та сенсорні пристрої є джерелами випадковості, особливо щодо молодших бітів, витягнутих із виміряних даних. Отримані результати показали, що можна отримати хороші значення ентропії, особливо для датчиків температури та світла. Крім того, другий експеримент, у якому датчики були запечатані в чорному ящику, виявлено, що ентропія залежить від

електронних пристроїв, за допомогою яких проводяться вимірювання. Експерименти показали високі значення ентропії, отримані від високо інерційних датчиків, таких як температура та освітлення, під час проведення на малих зразках.

1.2.5 Датчик гіроскопа

Датчик гіроскопа відповідає за вимірювання швидкості обертання навколо осі пристрою [10]. Однією з найкращих реалізацій гіроскопа є можливість плавного обертання та виконання кількох команд в іграх за допомогою 3D-рухів. На рис. 1.5 представлений принцип використання датчика гіроскопа.

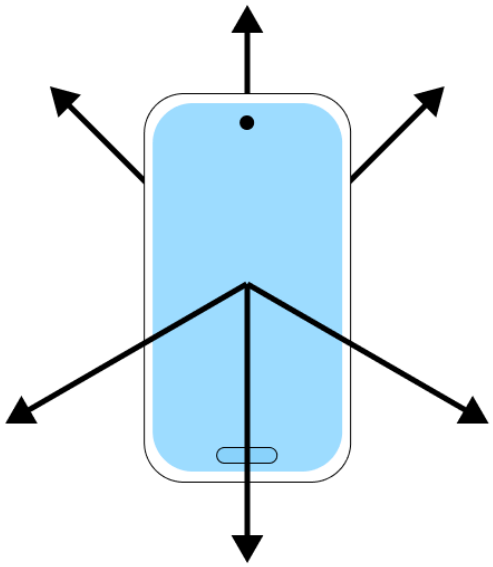

Рисунок 1.5 – Принцип використання датчика гіроскопа

У роботі [17] Seong-Min Cho та ін. запропонували генератор випадкових чисел, спеціально розроблений для безпілотних літальних апаратів, із вихідним кодом, отриманим на основі датчика даних, який використовується для польоту – гіроскоп. Вони використовували лише біти з ентропією для заповнення генератора випадкових чисел. Для цього вони визначили положення бітів, які генерують ентропію, в залежності від стану дрона: нерухомий або політ. Вони також порівнюють класичні генератори випадкових чисел, що використовуються для дронів, із запропонованим рішенням.

1.2.6 Датчик барометра

Датчик барометра використовується для визначення тиску навколишнього середовища [15]. Наприклад, програма здоров'я на смартфонах також використовує ці датчики. Піднімаючись зі сходів або переходячи з рівня землі на рівень підлоги, кожна деталь точно визначається датчиком барометра, а дані надсилаються на GPS. На рис. 1.6 зображено принцип використання датчика барометра.

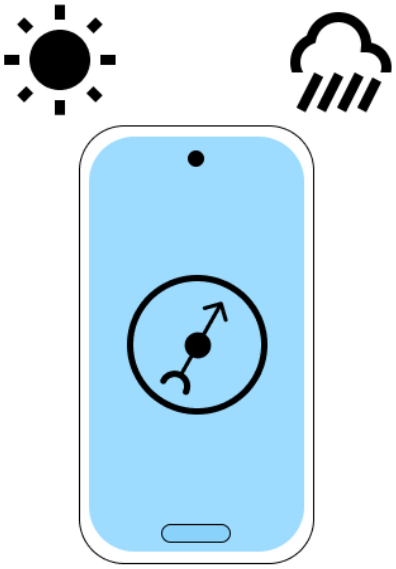

Рисунок 1.6 – Принцип використання датчика барометра

Також у роботі [17], яка згадувалася вище теж розглядали генератор випадкових чисел на основі датчиків барометру. Для отримання випадкових чисел використовували біти з ентропією.

1.2.7 Датчик магнітометра

Магнітометр – це датчик, який вимірює магнітне поле [13]. Існує два типи магнітометрів: стаціонарні, які використовуються для вимірювання у фіксованих точках, і мобільні, які використовуються в програмах, де потрібне виявлення руху. Беручи до уваги цей аспект, можна зрозуміти, чому датчики з достатньо високою чутливістю можуть генерувати ентропію. На рис. 1.7 зображено принцип використання датчика магнітометра.

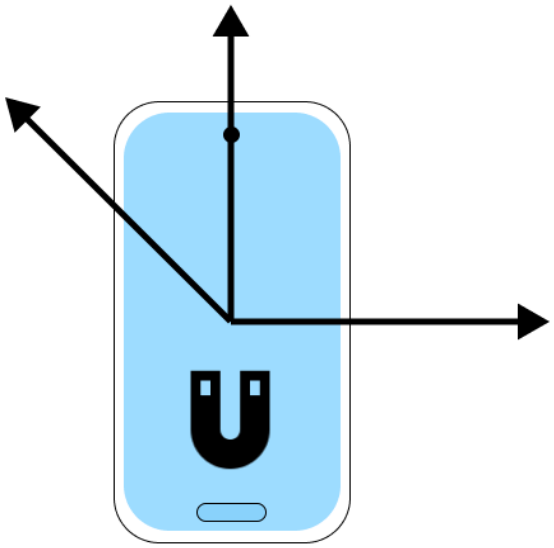

Рисунок 1.7 – Принцип використання датчика магнітометра

Важливим є придатність використання факторів навколишнього середовища, заснованих на мобільності та місці, як джерела ентропії в генераторі випадкових чисел. Прикладом є генерація пари ключів шифрування у потоці на основі умов навколишнього середовища, які пристрій може записати, таких як магнітне поле наближення, географічні координати та інші.

1.3 Вибір джерела ентропії для розробки генератора випадкових чисел на базі мобільного пристрою

При виборі джерела ентропії для розробки генератора випадкових чисел на базі мобільного пристрою, на думку автора, треба врахувати такі вимоги:

- чутливість датчика;
- наявність датчика у більшості мобільних пристроїв;
- швидкість оцифрування даних;
- кількість отриманих біт за одне вимірювання.

Порівняння датчиків смартфона у якості джерел ентропії приведено у таблиці 1.1.

Таблиця 1.1 – Порівняння датчиків смартфона у якості джерел ентропії

Назва датчика	Тип отримуваної інформації	Кількість біт на одне вимірювання	Доступність в пристроях	Відносна чутливість датчика
Акселерометр	Прискорення в 3 осях (x, y, z)	16-32 для кожної вісі	+	Висока
Датчик освітлення	Інтенсивність освітлення (люкс)	8-16	+	Середня
Гіроскоп	Кутові швидкості в 3 осях (x, y, z)	16-32 для кожної вісі	+	Висока
Барометр	Атмосферний тиск (бар)	16-32	Не в усіх пристроях	Низька
Магнітометр	Магнітне поле в 3 осях (x, y, z)	16-32 для кожної вісі	+	Середня

Акселерометр, гіроскоп та магнітометр наявні майже у всіх сучасних смартфонах, є досить чутливими, а також дають можливість отримання найбільшої кількості біт тому, що фіксують зміни у трьох координатах, тоді як датчик освітлення та барометр фіксують лише одне значення. Тому саме ці датчики було обрано в якості джерела ентропії для генератора випадкових чисел.

1.4 Висновки за розділом

Наведено основні поняття у сфері генерації випадкових чисел. Розглянуто основні джерела ентропії у мобільних пристроях, а саме: акселерометр, датчик положення, датчик освітлення, датчик гіроскопа, датчик барометра, датчик магнітометра. Проведено порівняльний аналіз датчиків та обрано акселерометр, гіроскоп і магнітометр для розробки генератора випадкових чисел.

2 АРХІТЕКТУРА, ФУНКЦІЇ ТА РЕЖИМИ РОБОТИ КОМПЛЕКСУ

2.1 Технічне завдання на розробку

Для подальшої можливості проведення досліджень в роботі передбачається створення апаратно-програмного комплексу для генерації випадкових та псевдовипадкових чисел з використанням джерел ентропії, що доступні в мобільних пристроях.

Склад розроблюваного комплексу:

- мобільний пристрій та фонове програмне забезпечення для нього;
- програмне забезпечення для сервера;
- приклад (шаблон) програмного забезпечення для клієнта.
- пристрій генерування випадкових чисел на базі мікроконтролера (розроблено в іншій частині комплексної роботи).

Режими генерування послідовностей чисел:

- генерація істинно випадкових 32-х бітних беззнакових цілих чисел за допомогою джерел ентропії, що доступні в мобільному пристрої;
- генерація псевдовипадкових 32-х бітних беззнакових цілих чисел, з використанням апаратно згенерованого зерна.

Вимоги до мобільного пристрою:

- наявність акселерометра, гіроскопа та магнітометра;
- наявність WiFi модулю.

Вимоги до серверного програмного забезпечення:

- можливість взаємодіяти одразу з декількома клієнтами;
- протокол має дозволяти встановлювати режим генерації (випадкові/псевдовипадкові числа);
- протокол має дозволяти встановлювати джерело ентропії (акселерометр/гіроскоп/магнітометр);
- протокол має дозволяти генерувати зерно для генератора псевдовипадкових чисел;

- протокол має дозволяти генерувати випадкові 32-х бітні беззнакові цілі числа по вказаному модулю.

Вимоги до клієнтського програмного забезпечення:

- можливість зчитування, формування і відправки команд сервера;
- можливість отримання відповідей від сервера і вивід їх на екран.

2.2 Архітектура комплексу

В даній роботі передбачається реалізувати апаратно-програмний комплекс який включає в себе генератор випадкових чисел на основі мобільного пристрою та серверу, що надає клієнтам зручний програмний інтерфейс для генерації випадкових чисел.

Як зазначено в підрозділі 2.1, комплекс має генерувати послідовності випадкових та псевдовипадкових чисел.

Генератор випадкових чисел – механізм, що претендує на генерацію справді випадкових чисел. Генератор випадкових чисел (Random Number Generator, RNG) використовує джерело ентропії, а також деяку функцію обробки для створення випадковості. Джерело ентропії зазвичай складається з деякої вимірюваної фізичної величини [7].

Генератор псевдовипадкових чисел (Pseudorandom Number Generator, PRNG) – це детермінований алгоритм, який, маючи на вході справжню випадкову бінарну послідовність довжиною k, виводить бінарну послідовність довжиною $l>>k$, яка виглядає випадковою. Вхід до генератора називається зерном (seed), а вихід – псевдовипадковою бітовою послідовністю (pseudorandom bit sequence). Генератор псевдовипадкових чисел використовує один або декілька вхідних параметрів для створення низки «псевдовипадкових» чисел. У випадках, коли потрібна максимальна непередбачуваність, початкове значення (зерно) повинно бути випадковим і непередбачуваним. Тому, зазвичай, PRNG має отримувати своє зерно від генератора випадкових чисел (RNG) [7].

Виходи PRNG зазвичай є детермінованими функціями зерна, що означає, що вся справжня випадковість обмежується лише процесом генерації цього зерна. Детермінований характер цього процесу обумовлює термін «псевдовипадковість». Оскільки кожен елемент псевдовипадкової послідовності можна відтворити за допомогою початкового зерна, для відтворення послідовності необхідно зберігати лише зерно [7].

Узагальнена архітектура комплексу зображена на рис. 2.1.

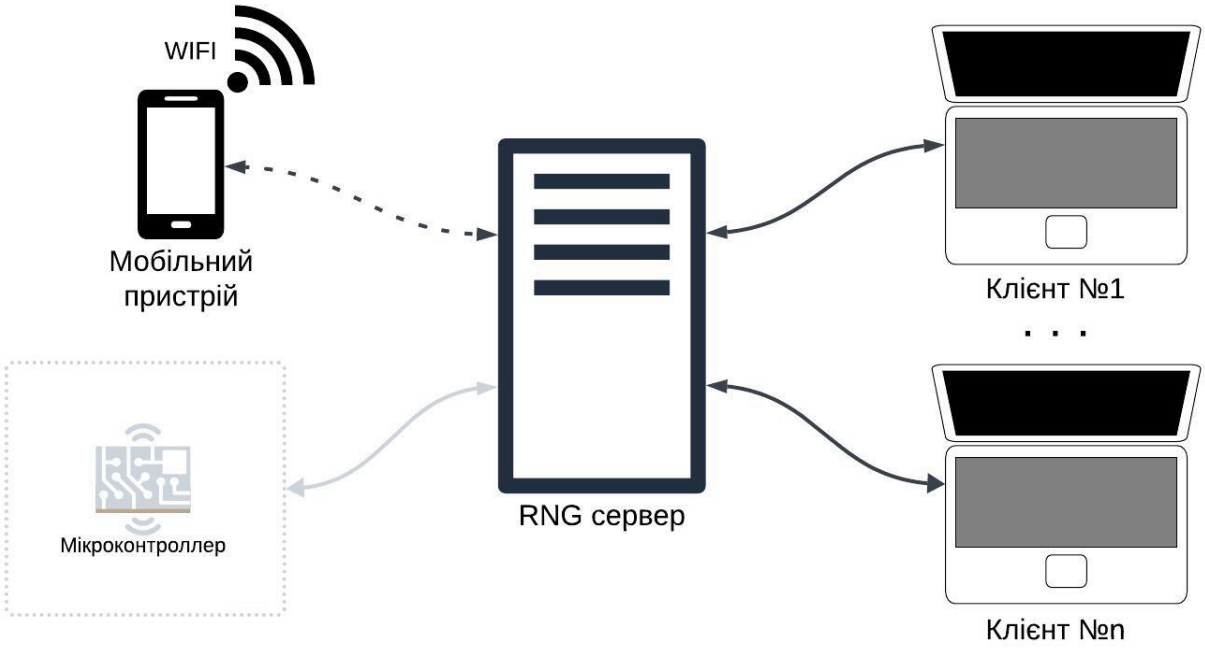

Рисунок 2.1 – Узагальнена архітектура комплексу

До складу комплексу (див. рис 2.1) також входить апаратний пристрій генерування випадкових чисел на базі мікроконтролера з використанням шумів напівпровідникових приладів, який розроблено в іншій частині комплексної роботи.

У якості протоколу обміну даних між мобільним пристроєм та сервером, і між сервером та клієнтами було обрано TCP. Протокол TCP був спеціально розроблений для надійної передачі потоку байтів між вузлами ненадійної мережі [18].

Серед переваг TCP-протоколу виділяють [18]:

- надійність: протокол гарантує, що всі пакети даних будуть доставлені без помилок і у правильному порядку;
- підтвердження доставки: кожен доставлений пакет даних підтверджується отримувачем;
- встановлення та розірвання з'єднання: TCP встановлює і закриває з'єднання перед початком та після завершення взаємодії відповідно.

Недоліки протоколу [18]:

- вища затримка: через необхідність встановлення з'єднання, а також підтвердження доставки і контроль помилок, протокол має вищу затримку ніж, наприклад, UDP;
- накладні витрати: використання додаткових заголовків для контролю передачі пакетів збільшує об'єм переданих даних, а також потребує більше ресурсів для обробки;
- не підходить для використання в реальному часі: необхідність підтвердження і контролю потоку даних робить TCP не найкращим варіантом для застосунків, де за рахунок можливих витрат важлива швидкість доставки даних.

Детальніша структура і зв'язки між елементами комплексу зображені на рис. 2.2.

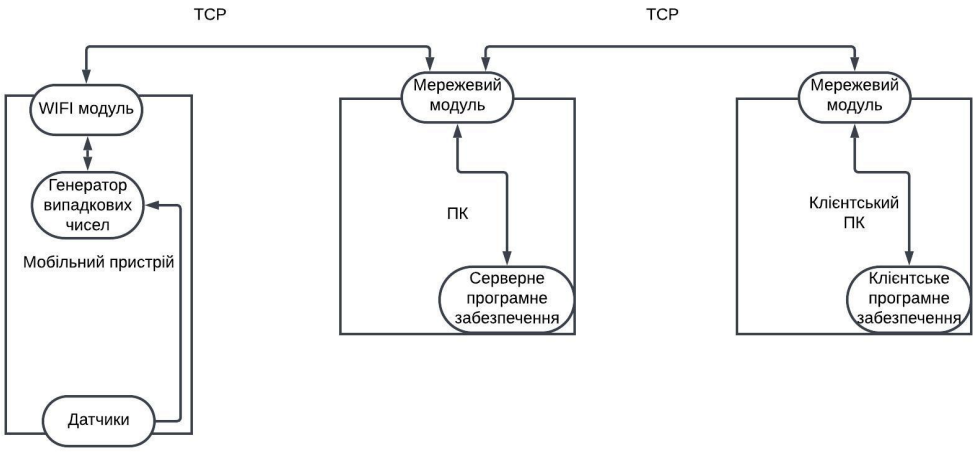

Рисунок 2.2 – Зв'язки між елементами комплексу

У якості мобільного пристрою було обрано пристрій під керуванням сімейства операційних систем iOS. У якості датчиків було обрано акселерометр, гіроскоп та магнітометр. Майже у всіх пристроях під керуванням iOS є ці датчики.

У якості алгоритму генерації псевдовипадкових чисел обрано BBS, через його математичну надійність (алгоритм ґрунтується на проблемі факторизації великих чисел, яка вважається складною для розв'язання) і високу ентропію (алгоритм гарантує, що згенеровані числа є максимально випадковими на практиці).

Алгоритм Блум-Блум-Шуба (англ. Algorithm Blum-Blum-Shub, BBS) – це криптографічний генератор псевдовипадкових чисел, що був розроблений на основі складності задачі факторизації. Алгоритм був запропонований у 1986 році Ленором Блумом, Мануелем Блумом і Майклом Шубом [19].

Формула генерації послідовності має наступний вигляд:

$$x_{n+1} = x_n^2 \, mod \, M, \tag{2.1}$$

де $M = pq$ є добутком двох великих простих чисел p і q, а з кожного x_{n+1} отримують наймолодший біт, який стає кожним наступним бітом випадкового числа.

Одною з вимог алгоритму є те, що p та q мають бути конгруентними з 3 по модулю 4. Також, чим менше найбільший спільний дільник НСД *(p-1,q-1)*, тим більше період послідовності. Одним з прикладів може виступати пара *p=7604, q=7487*.

Серед переваг алгоритму можна виділити:

- криптографічну стійкість, оскільки BBS базується на задачі факторизації великих чисел, яка є викликом навіть для сучасних комп'ютерів;
- простоту реалізації;

Однак, алгоритм також має декілька недоліків. Наприклад:

- невисока швидкість через необхідність виконання складних математичних операцій;
- залежність від великих чисел, що може вимагати значних ресурсів для їх генерації та зберігання;
- чутливість до початкового зерна, що потенційно може значно зменшити період псевдовипадкової послідовності.

2.3 Протоколи обміну даними між елементами комплексу

Для реалізації обміну даними між елементами комплексу необхідно розробити протоколии, що визначить правила і стандарти за якими елементи даного комплексу зможуть комунікувати із забезпеченням коректності і продуктивності даного процесу.

Як було вказано в технічному завданні, сервер повинен обробляти 4 типи команд від клієнта: команду встановлення режиму генерації (разом із полем, що ідентифікує режим), команду встановлення джерела ентропії (разом із полем, що ідентифікує джерело ентропії), команду генерації зерна для генератора псевдовипадкових чисел і команду генерації випадкового числа по модулю (разом із полем, що вказує модуль). А у якості реакції на ці команди передбачається 4 типи відповідей. Серед них два типи помилок: помилка, що вказує на неправильний формат наданої команди і помилка, що вказує на відсутність з’єднання з мобільний пристроєм. Також два типи відповідей успішного виконання команди: відповідь, що повідомляє про успішне виконання команди (наприклад, встановлення режиму генерації, встановлення джерела ентропії, або генерацію зерна), та відповідь, що повертає згенероване випадкове число.

Структуру запитів на генерацію випадкових чисел між клієнтом і сервером приведено у табл. 2.1.

Таблиця 2.1 – Структура запитів на генерацію випадкових чисел між клієнтом і сервером

Назва та інтерпретація запиту	Ідентифікатор команди	Можливі параметри	
Встановлення режиму генерації	SM (set mode)	Параметр 1 – ідентифікатор типу генератора: 0 – генератор випадкових чисел, 1 – генератор псевдовипадкових чисел	Параметр 2 – ідентифікатор джерела ентропії: 0 – акселерометр, 1 – гіроскоп, 2 – магнітометр
Генерація випадкового числа	GR (generate random)	модуль (від 1 до $2^{32}-1$)	

Виходячи з даних табл. 2.1 можна зробити такі запити до сервера:

- SM00 – встановити режим генерації випадкових чисел за допомогою акселерометра;
- SM01 – встановити режим генерації випадкових чисел за допомогою гіроскопа;
- SM02 – встановити режим генерації випадкових чисел за допомогою магнітометра;
- SM10 – встановити режим генерації псевдовипадкових чисел з зерном згенерованим за допомогою акселерометра;
- SM11 – встановити режим генерації псевдовипадкових чисел з зерном згенерованим за допомогою гіроскопа;

- SM12 – встановити режим генерації псевдовипадкових чисел з зерном згенерованим за допомогою магнітометра;
- GR100 – генерація випадкового числа по модулю 100, використовуючи встановлений режим.

Структуру відповідей сервера клієнту приведено у табл. 2.2.

Таблиця 2.2 – Структура відповідей сервера клієнту

Інтерпретація відповіді	Зміст відповіді
Число успішно згенеровано	OK[згенероване число]
Режим генерації встановлено	OK
Неправильний формат запиту	EF
Мобільний пристрій не підключений	EC

Виходячи з даних табл. 2.2 можна отримати такі відповіді від сервера:

- OK10 – згенероване випадкове число;
- OK – режим генерації було встановлено;
- EF – сервер отримав неправильно сформований запит;
- EC – мобільний пристрій не з’єднаний з сервером.

Для реалізації обміну даних між сервером і мобільним пристроєм передбачається лише один тип запиту, де вказано ідентифікатор джерела ентропії. Наприклад:

- 0 – генерація випадкового числа за допомогою акселерометра;
- 1 – генерація випадкового числа за допомогою гіроскопа;
- 2 – генерація випадкового числа за допомогою магнітометра.

Натомість мобільний пристрій відповідає згенерованим числом.

На рис. 2.3 та на рис. 2.4 зображено приклади діаграм обміну повідомленнями (запитів та відповідей). Кроки роботи протоколу обміну показані в дужках.

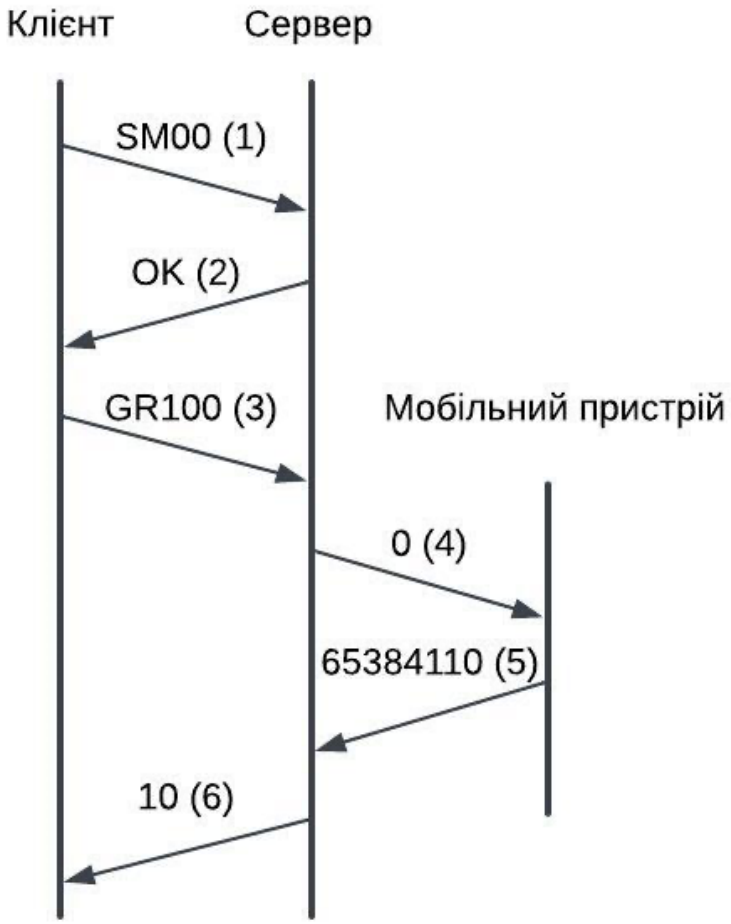

Рисунок 2.3 – Приклад діаграми запитів і відповідей для генерації випадкового числа

Пояснення до рис. 2.3.

Крок 1 – відправка клієнтом до сервера запиту на встановлення режиму генерації випадкових чисел за допомогою акселерометра.

Крок 2 – відправка сервером до клієнта відповіді про успішне встановлення режиму.

Крок 3 – відправка клієнтом до сервера команди на генерацію випадкового числа по модулю 100.

Крок 4 – відправка сервером команди до мобільного пристрою на генерацію випадкового числа за допомогою акселерометру.

Крок 5 – відправка мобільним пристроєм згенерованого випадкового числа до серверу.

Крок 6 – відправка сервером клієнту згенерованого випадкового числа по модулю 100.

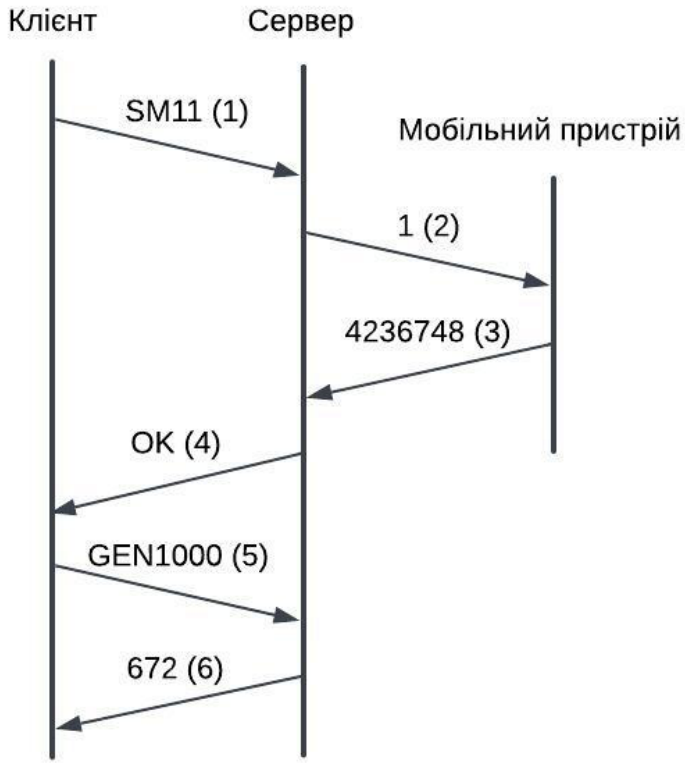

Рисунок 2.4 – Приклад діаграми запитів і відповідей для генерації псевдовипадкового числа

Пояснення до рис. 2.4.

Крок 1 – відправка клієнтом до сервера запиту на встановлення режиму генерації псевдовипадкових чисел із зерном згенерованим за допомогою гіроскопа.

Крок 2 – відправка сервером команди до мобільного пристрою на генерацію випадкового числа за допомогою гіроскопа.

Крок 3 – відправка мобільним пристроєм згенерованого випадкового числа до сервера і встановлення його у якості зерна для генератора псевдовипадкових чисел.

Крок 4 – відправка сервером до клієнта відповіді про успішне встановлення режиму.

Крок 5 – відправка клієнтом до сервера команди на генерацію випадкового числа по модулю 1000.

Крок 6 – відправка сервером клієнту згенерованого випадкового числа по модулю 1000.

2.4 Висновки за розділом

Сформовано технічне завдання на розробку апаратно-програмного комплексу. Наведено архітектуру комплексу та зв’язки між його елементами. В якості алгоритму генерації псевдовипадкових чисел обрано BBS. Для взаємодії між елементами комплексу обрано протокол TCP. Розроблено систему команд для взаємодії між елементами комплексу.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМПЛЕКСУ

3.1 Вибір засобів і середовищ розробки

Одним із головних етапів розробки програмного забезпечення є вибір інструментів для розробки, що визначатиме ефективність, зручність і швидкість реалізації проекту. Правильний вибір значно полегшить процес розробки і тестування, а також дозволить значно скоротити час розробки та супровід програми. Враховуючи специфіку вимог до проекту, чи не найважливіше обрати такі технології, які оптимально відповідатимуть характеристикам майбутнього програмного забезпечення.

У цьому підрозділі розглядається процес вибору мов програмування, фреймворків і середовищ розробки, які були застосовані в рамках реалізації даної роботи. Вибір технологій базується на таких критеріях, як продуктивність, швидкодія, доступність необхідних бібліотек, зручність для автора так інших факторах.

Для реалізації серверної і клієнтської частини комплексу обрано мову програмування Go [20].

Основні переваги мови [20]:

- проста для вивчення;
- кросплатформовість;
- мова з відкритим кодом, яка підтримується Google;
- вбудований паралелізм;
- всебічно розроблена стандартна бібліотека;
- зручність для створення клієнт-серверних застосунків;
- велика екосистема бібліотек, інструментів та широка спільнота розробників.

У якості середовища розробки для серверної та клієнтської частини обрано JetBrains Goland [21]. Скріншот середовища представлено на рис. 3.1.

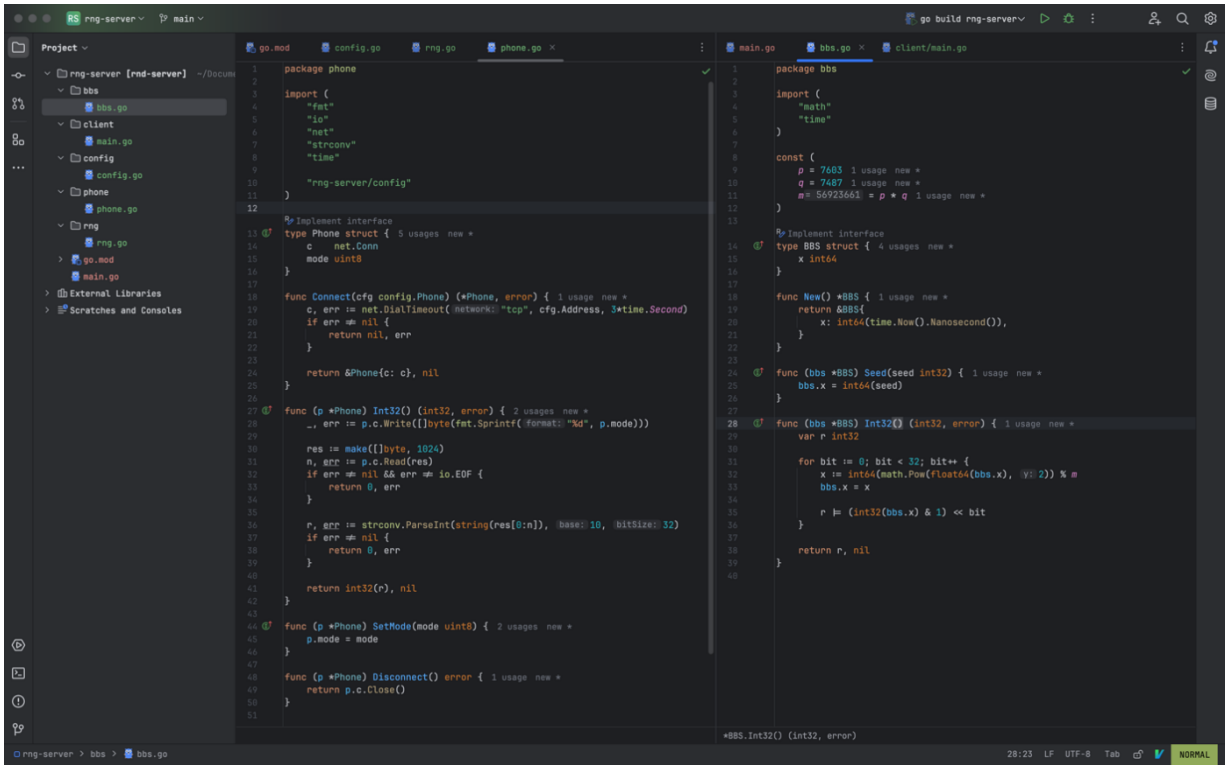

Рисунок 3.1 – Середовище розробки JetBrains Goland

Середовище має такі переваги [21]:

- продуктивність;
- відмінне автодоповнення та вбудований штучний інтелект;
- зручний дебагер;
- нативна підтримка інших суміжних технологій.

Для реалізації програмного забезпечення для мобільного пристрою було обрано мову Dart, створену для зручної розробки користувацьких інтерфейсів [22] разом із фреймворком Flutter для розробки під мобільні операційні системи [23].

Переваги фреймворку включають [23]:

- кросплатформовість;
- швидкість розробки;
- швидкодія;
- єдиний код для всіх платформ;
- багатий набір готових віджетів;

- велика екосистема бібліотек.

У якості середовища розробки для мобільного пристрою обрано Android Studio [24] разом з плагіном для розробки програмного забезпечення з використанням фреймворку Flutter. Скріншот середовища представлено на рис. 3.2.

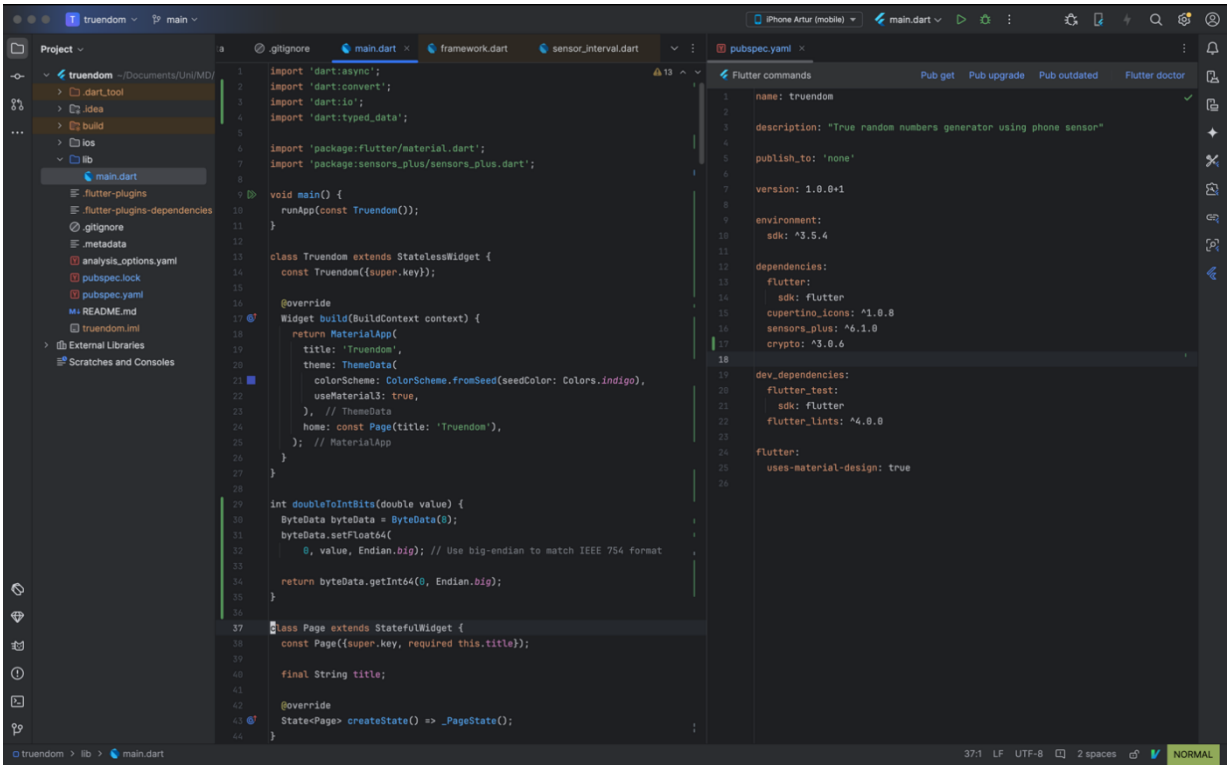

Рисунок 3.2 – Середовище розробки Android Studio

До переваг середовища входить [24]:

- відмінне автодоповнення та вбудований штучний інтелект;
- зручне створення користувацьких інтерфейсів;
- наявність плагіну для розробки використовуючи фреймворк Flutter.

У роботі використовується Flutter бібліотека «sensors_plus» [25] для взаємодії з датчиками мобільного пристрою.

3.2 Розробка програмного забезпечення серверної частини

Для серверної частини комплексу розроблено програмне забезпечення використовуючи мову Go та середовище розробки JetBrains Goland. Код

програми наведено в додатку А. На рис. 3.3 приведено узагальнений алгоритм програми.

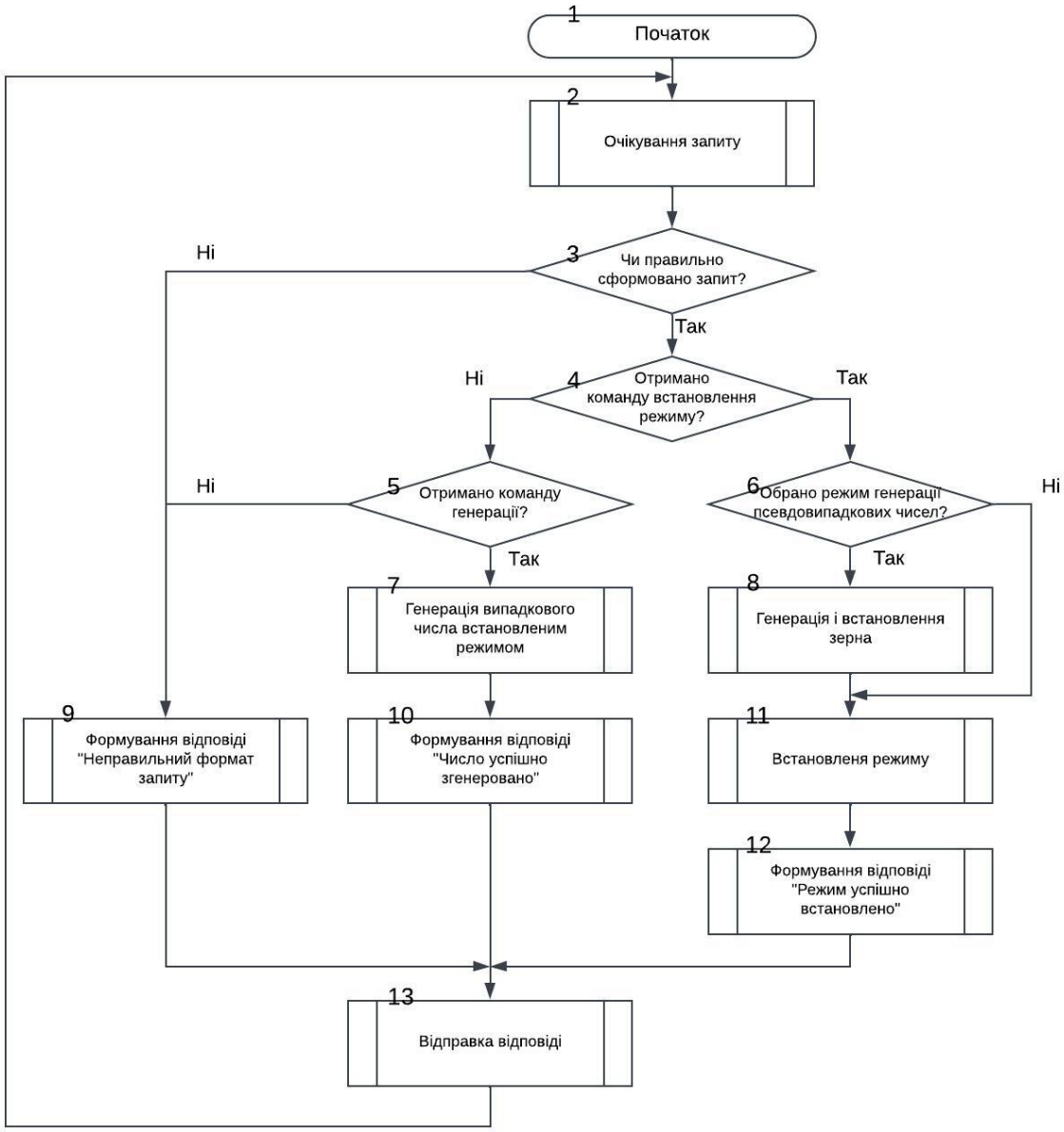

Рисунок 3.3 – Узагальнений алгоритм обробки сервером запиту від клієнта

Блок 1: початок.

Блок 2: очікування запиту клієнта.

Блок 3: валідація запиту.

Блок 4: перевірка чи отримано команду на встановлення режиму генерації.

Блок 5: перевірка чи отримано команду на генерацію випадкового числа.

Блок 6: перевірка чи обрано режим генерації псевдовипадкових чисел.

Блок 7: генерація випадкового числа встановленим режимом.

Блок 8: генерація і встановлення зерна для генератора псевдовипадкових чисел.

Блок 9: формування відповіді про неправильний формат запиту.

Блок 10: формування відповіді про успішну генерацію випадкового числа.

Блок 11: встановлення режиму генерації.

Блок 12: формування відповіді про успішно встановлений режим.

Блок 13: відправка відповіді клієнту.

3.3 Розробка програмного забезпечення мобільного пристрою

Для мобільного пристрою розроблено фонове програмне забезпечення використовуючи мову Dart, фреймворк для розробки мобільних застосунків Flutter та середовище розробки Android Studio. Код програми наведено в додатку Б. На рис. 3.4 приведено узагальнений алгоритм програми.

Рисунок 3.4 – Узагальнений алгоритм роботи програми мобільного пристрою

Блок 1: запуск програми.

Блок 2: очікування на запит сервера.

Блок 3: встановлення режиму генерації (акселерометр, гіроскоп чи магнітометр).

Блок 4: генерація випадкового числа використовуючи встановлений режим.

Блок 5: відправка серверу згенерованого випадкового числа.

3.4 Розробка прикладу програмного забезпечення клієнтської частини

Для прикладу клієнтського застосунку розроблено просту консольну програму, яка встановлює TCP-з'єднання з сервером і дає можливість виконувати команди користувача введені з клавіатури. Код програми наведений у додатку В. Клієнтський застосунок необхідний для подальших досліджень. Алгоритм прикладу програмного забезпечення клієнтської частини приведений на рис. 3.5.

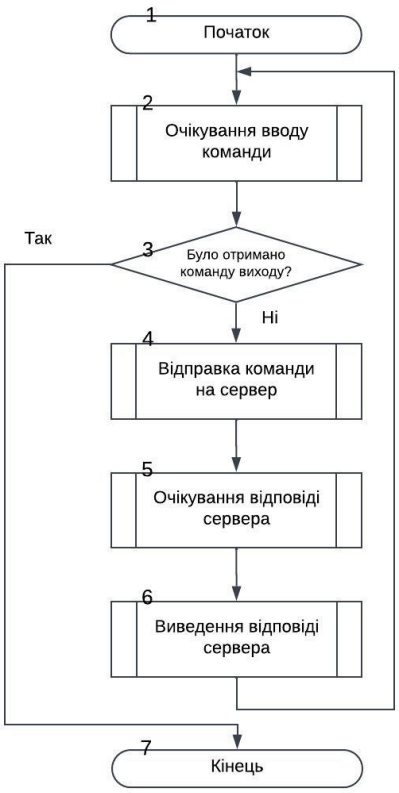

Рисунок 3.5 – Узагальнений алгоритм роботи прикладу клієнтської частини

3.5 Налагодження та перевірка працездатності комплексу

Для налагодження розробленого програмного комплексу можна скористатись наступною інструкцією.

Для запуску серверної програми можна використати команду «go run main.go». Для її зупинки використовується комбінація клавіш Ctrl+C. Програма зупиниться тільки коли всі клієнти від’єднаються від сервера.

Для запуску клієнтської програми можна використати команду «SERVER_ADDRESS=localhost:8080 go run client/main.go», де

- SERVER_ADDRESS – IP-адреса (включаючи порт) серверу

Після запуску клієнтська програма очікує на введення запитів.

На рис. 3.6-3.10 наведено результати налагодження роботи комплексу.

```
aopry@D9V4MX6CY5 rng-server % go run main.go
{"time":"2025-01-16T16:18:57.128279+02:00","level":"INFO","msg":"RNG server"}
{"time":"2025-01-16T16:18:57.129087+02:00","level":"INFO","msg":"server started"}
^C{"time":"2025-01-16T16:20:16.497881+02:00","level":"INFO","msg":"server closed"}
aopry@D9V4MX6CY5 rng-server %
```

Рисунок 3.6 – Запуск та завершення роботи сервера

```
aopry@D9V4MX6CY5 rng-server % SERVER_ADDRESS=localhost:8080 go run client/main.go
RNG client
Enter a command: SM00
Response: OK
Enter a command: GR1000
Response: OK465
Enter a command: GR1000
Response: OK201
Enter a command: GR1000
Response: OK530
Enter a command: GR1000
Response: OK403
Enter a command: GR1000
Response: OK370
Enter a command: ^Csignal: interrupt
aopry@D9V4MX6CY5 rng-server %
```

Рисунок 3.7 – Приклад запитів клієнта на генерацію випадкових чисел за допомогою акселерометра

```
aopry@D9V4MX6CY5 rng-server % SERVER_ADDRESS=localhost:8080 go run client/main.go
RNG client
Enter a command: SM11
Response: OK
Enter a command: GR1000
Response: OK575
Enter a command: GR1000
Response: OK320
Enter a command: GR1000
Response: OK19
Enter a command: GR1000
Response: OK669
Enter a command: GR1000
Response: OK465
Enter a command:
```

Рисунок 3.8 – Приклад запитів клієнта на генерацію псевдовипадкових чисел із зерном згенерованим за допомогою гіроскопа

```
aopry@D9V4MX6CY5 rng-server % SERVER_ADDRESS=localhost:8080 go run client/main.go
RNG client
Enter a command: SM23
Response: EF
Enter a command: ^Csignal: interrupt
aopry@D9V4MX6CY5 rng-server %
```

Рисунок 3.9 – Приклад отримання клієнтом помилки про неправильно сформований запит

```
aopry@D9V4MX6CY5 rng-server % SERVER_ADDRESS=localhost:8080 go run client/main.go
RNG client
Enter a command: SM11
Response: EC
Enter a command: ^Csignal: interrupt
aopry@D9V4MX6CY5 rng-server %
```

Рисунок 3.10 – Приклад отримання клієнтом помилки про відсутність з’єднання з мобільним пристроєм

3.6 Висновки за розділом

Обрано Go в якості мови програмування та JetBrains Goland у якості середовища розробки для створення серверної та клієнтської частин комплексу. Dart обрано у якості мови програмування, Flutter у якості фреймворку для

розробки під мобільні операційні системи та Android Studio у якості середовища розробки для створення програмного забезпечення для мобільного пристрою. Розроблені блок-схеми узагальнених алгоритмів роботи основних частин комплексу. Написане програмне забезпечення для всіх частин комплексу. Налагоджено роботу комплексу та перевірено його працездатність.

4 ДОСЛІДЖЕННЯ ЯКОСТІ ВИПАДКОВИХ ЧИСЕЛ, ОТРИМАНИХ ЗА ДОПОМОГОЮ КОМПЛЕКСУ

4.1 Засоби для дослідження ступеню випадковості послідовностей чисел

Серед можливих засобів перевірки послідовностей чисел на випадковість в роботі прийнято рішення використовувати кейс тестів NIST [7] та візуальний графічний тест [26].

Кейс тестів NIST складається з 15 тестів, які були розроблені для перевірки на випадковість двійкові послідовності довільної довжини, згенеровані за допомогою апаратних або програмних випадкових та псевдовипадкових генераторів. Кейс тестів включає [7]:

- The Frequence (Monobit) Test (частотний тест) – метою цього тесту є перевірка, чи є кількість одиниць і нулів у послідовності такою ж якою можна очікувати від ідеальної випадкової послідовності. Тобто, в ідеальній випадковій послідовності повинно бути приблизно стільки ж одиничних бітів, скільки й нульових.

- Frequency Test within a Block (тест частоти в середині блоку) – метою цього тесту є перевірка що кількість одиничних бітів в блоці встановленої довжини однаковою з кількістю нульових біт, тобто чи правильно розподілено одиниці і нулі в межах блоків послідовності фіксованої довжини.

- The Runs Test (тест на пробіги) – тест, який оцінює випадковість послідовностей біт, перевіряючи, чи є чергування одиниць і нулів занадто довгими, або навпаки занадто короткими. Цей тест допомагає визначити, чи існує надмірна або недостатня кількість змін стану (з 0 на 1 чи навпаки).

- Tests for the Longest-Run-of-Ones in a Block (тест на найдовший пробіг одиниць у блоці) – тест, що перевіряє, чи є у послідовності надмірно довгі серії бітів у блоках встановленої довжини.

- The Binary Matrix Rank Test (тест рангу двійкової матриці) – тест оцінює, чи є матриця, сформована з послідовності бітів, лінійно незалежною.

Використовується для виявлення кореляцій або структур у даних, що можуть вказувати на не випадковий характер послідовності.

- The Discrete Fourier Transform (Spectral) Test (тест дискретного перетворення Фур'є, або спектральний тест) – базується на аналізі частотного спектру послідовності та дозволяє виявити, чи існують у послідовності регулярні патерни, що можуть вказувати на її невипадковість.

- The Non-overlapping Template Matching Test (тест на перехресне зіставлення шаблонів) – перевіряє, чи зустрічаються у послідовності певні встановлені шаблони, які можуть вказувати на закономірності в даних.

- The Overlapping Template Matching Test (тест на перекривне зіставлення шаблонів) – перевіряє, чи зустрічаються в послідовності певні встановлені шаблони, дозволяючи шаблонам перекриватися (тобто одна частина шаблону може бути спільною з іншою при пошуку). Це дозволяє більш точно оцінити регулярності у послідовностях.

- Maurer's "Universal Statistical" Test – ґрунтується на теорії інформації та використовує принципи теоретичної ентропії для визначення, чи є послідовність статистично випадковою або ж має властивості, що вказують на її непередбачуваність.

- The Linear Complexity Test (тест на лінійну складність) – визначає наскільки складною є послідовність з точки зору лінійних рекурентних співвідношень. Вона вимірює, чи можна відтворити послідовність за допомогою лінійного рекурентного співвідношення або ж її генерація потребує більш складних алгоритмів.

- The Serial Test (серійний тест) – оцінює, наскільки добре у послідовності розподіляються підпослідовності певної довжини. Зокрема, цей тест перевіряє пари, трійки бітів тощо, в залежності від параметрів тесту.

- The Approximate Entropy Test (тест приблизної ентропії) – як і серійний тест, увага в цьому тесті зосереджена на частоті всіх можливих перекриваючихся шаблонів встановленої довжини.

- The Cumulative Sums (Cusums) Test (тест кумулятивних сум) – використовує накопичені суми, щоб виявити будь-які статистичні аномалії або тенденції послідовності. Тест базується на ідеї, що у випадкових послідовностях значення накопичених сум повинні залишатися в межах певних статистичних меж.

- The Random Excursions Test (тест випадкових екскурсій) – використовується для перевірки поведінки кумулятивних сум. Оцінює наскільки часто і наскільки далеко кумулятивна сума відходить від 0, що дає змогу виявити закономірності.

- The Random Excursions Variant Test (тест варіантів випадкових екскурсій) – варіант класичного Random Excursions Test, але з додатковими змінними, що враховують певні варіанти і властивості екскурсій.

Для перевірки послідовностей за допомогою кейсу NIST обрано програму NIST Randomness Testsuite [26]. Скріншот програми приведено на рис. 4.1.

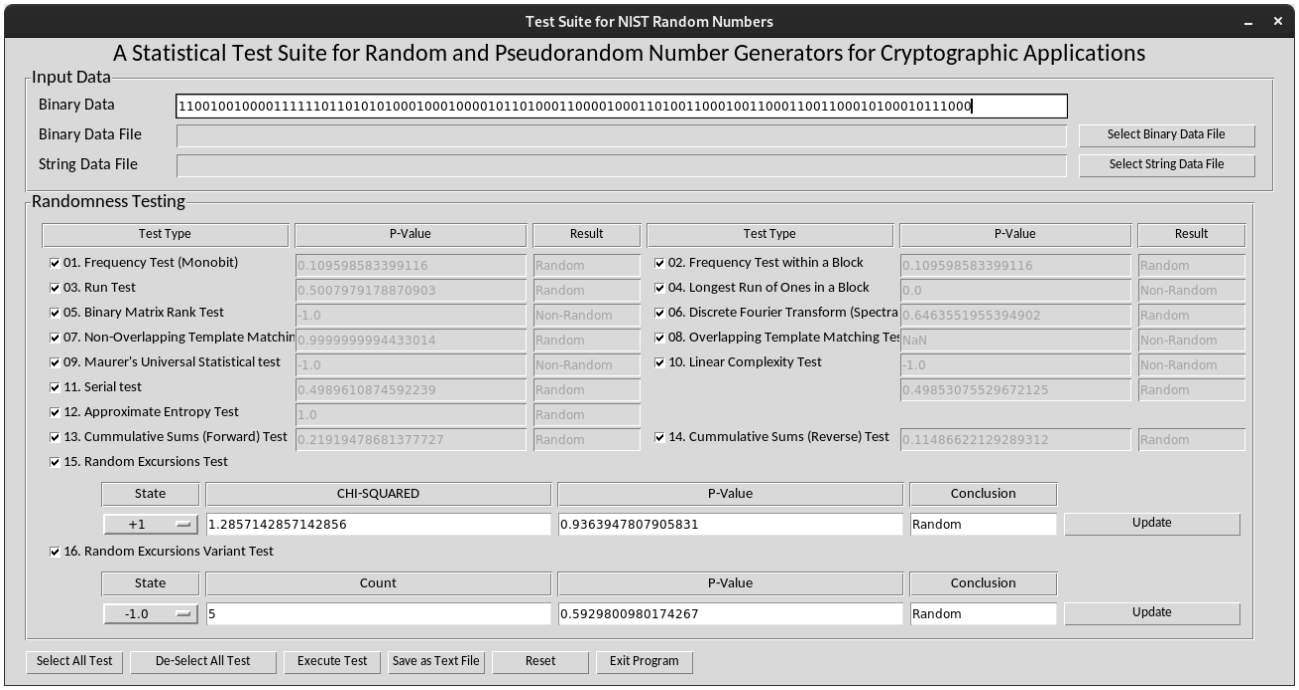

Рисунок 4.1 – Скріншот програми NIST Randomness Testsuite

У якості візуального графічного тесту пропонується генерувати зображення, в якому колір кожного наступного пікселя позначає біт випадкової послідовності.

Чорний колір позначає одиницю, а білий – 0. Якщо послідовність біт випадкова, то на зображенні не повинно бути видимих патернів, і картинка має виглядати як «шум» - хаотичне розміщення чорних і білих пікселів. Якщо ж є повторювані патерни, це може вказувати на те, що послідовність не є випадковою. Приклад результатів графічного тесту приведено на рис. 4.2 на 4.3 [26].

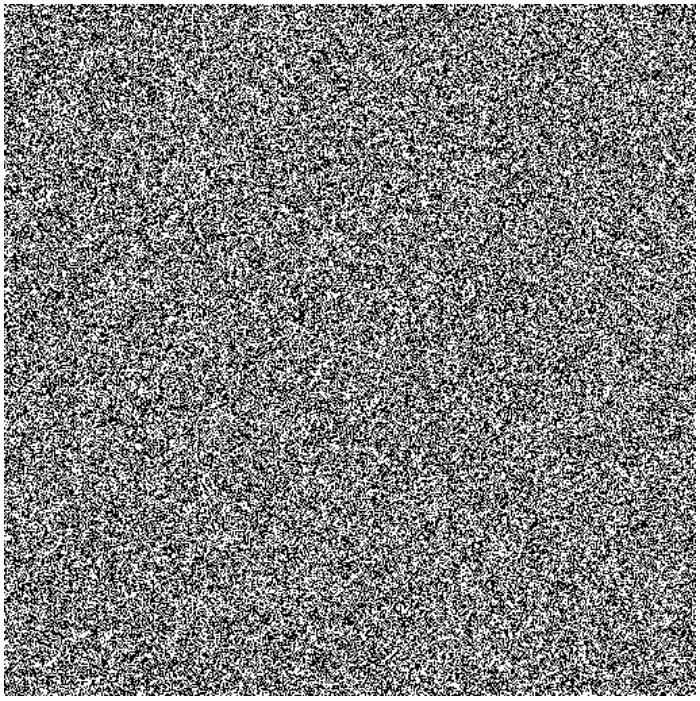

Рисунок 4.2 – Приклад результату графічного тесту без видимих патернів

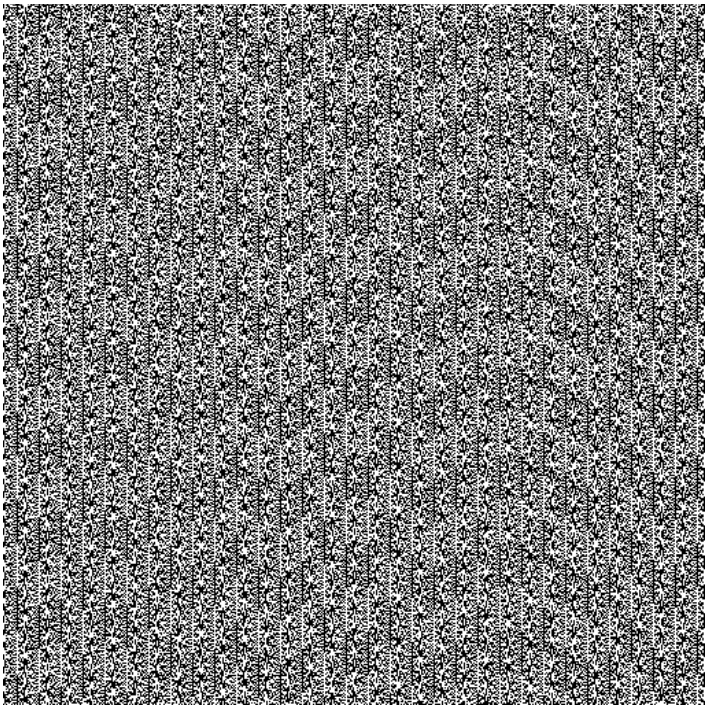

Рисунок 4.3 – Приклад результату графічного тесту з видимими патернами

Для проведення експериментів використовується лабораторний макет на базі MacBook Pro M3 (для запуску клієнт та сервера) та IPhone 14. Лабораторний макет зображено на рис. 4.4.

Рисунок 4.4 – Лабораторний макет

Технічні характеристики комп’ютера:

- процесор: Apple M3 Pro (12-core CPU, 18-core GPU);
- оперативна пам’ять: 36GB;
- операційна система: macOS Sequoia (version 15.2).

4.2 Визначення ефективного способу використання датчиків мобільного пристрою у якості джерел ентропії

У роботі розроблено спеціальний режим комплексу для перевірки якості згенерованих випадкових чисел, використовуючи різні способи їх отримання з використаних датчиків.

Приклади наборів даних, які можна отримати з використовуваних в роботі датчиків, наведено на рис. 4.5 – 4.7, відповідно.

```
x: 0.18052459716796876, y: 8.894503784179689, z: 4.1393421936035155
x: 0.19609222412109376, y: 8.922645263671875, z: 4.088597717285157
x: 0.16196319580078125, y: 8.918154602050782, z: 4.108955383300781
x: 0.17124389648437502, y: 8.890462188720704, z: 4.144281921386719
x: 0.22408401489257815, y: 8.895850982666015, z: 4.151167602539063
x: 0.16854949951171877, y: 8.896449737548828, z: 4.1323068237304685
x: 0.173638916015625, y: 8.90992172241211, z: 4.09593246459961
x: 0.204774169921875, y: 8.912466430664063, z: 4.110452270507813
x: 0.15657440185546875, y: 8.89465347290039, z: 4.143683166503907
x: 0.1730401611328125, y: 8.891809387207031, z: 4.134402465820313
x: 0.202379150390625, y: 8.905880126953125, z: 4.12661865234375
x: 0.195194091796875, y: 8.875193939208986, z: 4.119732971191406
x: 0.20492385864257814, y: 8.91321487426758, z: 4.12661865234375
x: 0.18411712646484377, y: 8.898545379638673, z: 4.133654022216797
x: 0.142503662109375, y: 8.893755340576172, z: 4.134552154541016
x: 0.2375559997558594, y: 8.900491333007812, z: 4.126319274902344
x: 0.22842498779296877, y: 8.89749755859375, z: 4.132007446289062
x: 0.16540603637695314, y: 8.893306274414062, z: 4.1433837890625
x: 0.18291961669921875, y: 8.900191955566406, z: 4.1411384582519535
x: 0.17693206787109375, y: 8.904832305908204, z: 4.118984527587891
x: 0.19953506469726565, y: 8.894803161621095, z: 4.135001220703125
x: 0.16046630859375, y: 8.961564331054689, z: 4.168082427978516
x: 0.18531463623046876, y: 8.90183853149414, z: 4.131708068847656
x: 0.18067428588867188, y: 8.902437286376953, z: 4.155508575439454
x: 0.16316070556640624, y: 8.900042266845704, z: 4.122127990722657
x: 0.25357269287109374, y: 8.91710678100586, z: 4.126768341064453
x: 0.1836680603027344, y: 8.906478881835938, z: 4.120930480957031
x: 0.13382171630859377, y: 8.892108764648437, z: 4.1288639831542975
x: 0.19908599853515627, y: 8.906478881835938, z: 4.132905578613282
x: 0.19773880004882813, y: 8.907826080322266, z: 4.112697601318359
x: 0.07200027465820313, y: 8.881331176757813, z: 4.130809936523438
```

Рисунок 4.5 – Прикладу набору даних акселерометра, виведених в консоль

```
x: 0.015908125787973404, y: -0.01739310473203659, z: 0.0044205812737345695
x: 0.009982593357563019, y: 0.011665711179375648, z: -0.0038836877793073654
x: -0.009955162182450294, y: -0.00221947836689651, z: 0.0022389194928109646
x: -0.015503591857850552, y: 0.007370031904429197, z: 0.002139317337423563
x: 0.00538464542478323, y: -0.0001536643976578489, z: 0.0004396878939587623
x: 0.009520001709461212, y: -0.0027627632953226566, z: -0.0009177253232337534
x: -0.004591023549437523, y: 0.002900981344282627, z: 0.0009571401169523597
x: -0.009457151405513287, y: 0.001350488979369402, z: 0.0013792511308565736
x: -0.0016519587952643633, y: -0.003114300547167659, z: -0.00023915186466183513
x: 0.004583033733069897, y: 0.0026402578223496675, z: -0.00031664984999224544
x: 0.0014367754338309169, y: 0.00330231967382133, z: 0.00003302319601061754
x: -0.00304638990201056, y: -0.0023249397054314613, z: 0.0027923244051635265
x: -0.005510612856596708, y: 0.002183259464800358, z: 0.0018959043081849813
x: 0.000999750685878098, y: 0.0020274645648896694, z: -0.0005941512063145638
x: 0.0010966897243633866, y: 0.0022418489679694176, z: 0.00022184131375979632
x: 0.0036605149507522583, y: -0.011503257788717747, z: 0.0007097324123606086
x: -0.006687996443361044, y: 0.00606428412720561, z: -0.0010250506456941366
x: -0.0010788465151563287, y: 0.0017800568602979183, z: 0.002293247962370515
x: -0.0010636665392667055, y: -0.009221461601555347, z: 0.0006719155353493989
x: -0.003510579001158476, y: 0.005775597412139177, z: 0.0006878945278003812
x: 0.0018024274613708258, y: 0.00001411475386703 3869, z: -0.0006940197781659663
x: -0.00012756542128045112, y: -0.005044825840741396, z: 0.002700977958738804
x: -0.002502306131646037, y: -0.0022213426418602467, z: 0.0005440838285721838
x: -0.0022037657909095287, y: 0.004354001954197884, z: 0.0012263857061043382
x: 0.0024913870729506016, y: -0.006743390113115311, z: 0.001026382320560515
x: 0.000639691308606416, y: -0.0031848743092268705, z: 0.001683650421909988
x: -0.004099137615412474, y: 0.001377386855892837, z: -0.00108257494866848
x: -0.0006500775925815105, y: -0.0014748586108908057, z: 0.0009030779474414885
x: -0.0000450074221589 6025, y: 0.0012141350889578462, z: -0.002130795270204544
x: 0.0012990899849683046, y: 0.0019244002178311348, z: 0.0018096179701387882
x: 0.0008295746520161629, y: -0.003561445279046893, z: 0.0010900318156927824
x: -0.0054128747433424, y: 0.004217648413032293, z: 0.0005765743553638458
x: -0.0024413196370005608, y: 0.0007416903390549123, z: 0.0007651261985301971
x: 0.001728657865896821, y: -0.0025009745731949806, z: 0.0018703379901126027
x: 0.002426938619464636, y: -0.00020266656065359712, z: -0.0008274441352114081
```

Рисунок 4.6 – Приклад набору даних гіроскопа, виведених в консоль

```
x: -128.0546875, y: 84.324462890625, z: -340.8043212890625
x: -127.80438232421875, y: 84.328369140625, z: -341.1653747558594
x: -128.00497436523438, y: 84.37646484375, z: -341.11334228515625
x: -128.17967224121094, y: 84.19140625, z: -341.33990478515625
x: -128.07577514648438, y: 84.26268005371094, z: -341.1424255371094
x: -128.00474548339844, y: 84.1646728515625, z: -340.4656677246094
x: -128.20718383789062, y: 84.25277709960938, z: -340.97406005859375
x: -128.3074493408203, y: 84.28596496582031, z: -340.713623046875
x: -127.97660827636719, y: 84.02854919433594, z: -340.84295654296875
x: -127.9393310546875, y: 84.14454650878906, z: -340.7763366699219
x: -127.95770263671875, y: 84.2901611328125, z: -340.8863830566406
x: -128.0269775390625, y: 84.32548522949219, z: -340.7537841796875
x: -127.99520874023438, y: 84.21224975585938, z: -340.6144104003906
x: -127.85246276855469, y: 84.17303466796875, z: -340.7722473144531
x: -127.81471252441406, y: 84.18498229980469, z: -340.73419189453125
x: -127.6375732421875, y: 84.16288757324219, z: -340.88330078125
x: -127.802490234375, y: 83.74592590332031, z: -340.68035888671875
x: -127.58900451660156, y: 83.77668762207031, z: -340.73309326171875
x: -127.75886535644531, y: 83.79244995117188, z: -340.77490234375
x: -127.94326782226562, y: 84.33335876464844, z: -340.8023376464844
x: -127.76980590820312, y: 84.12469482421875, z: -340.7876281738281
x: -127.92892456054688, y: 84.22486877441406, z: -340.74969482421875
x: -127.86001586914062, y: 84.38217163085938, z: -340.88818359375
x: -127.85853576660156, y: 84.38496398925781, z: -341.09490966796875
x: -127.696044921875, y: 84.22212219238281, z: -340.71636962890625
x: -127.52880859375, y: 84.44183349609375, z: -340.92279052734375
x: -127.90948486328125, y: 84.36882019042969, z: -340.9520263671875
x: -127.7181396484375, y: 84.42141723632812, z: -340.7213134765625
x: -127.915771484375, y: 84.28445434570312, z: -340.76824951171875
x: -127.95440673828125, y: 84.277587890625, z: -340.93475341796875
x: -128.01344299316406, y: 84.33558654785156, z: -340.76776123046875
x: -127.98686218261719, y: 84.48289489746094, z: -340.53289794921875
x: -128.1085662841797, y: 84.16972351074219, z: -340.6369934082031
x: -128.01165771484375, y: 84.63497924804688, z: -340.2728271484375
x: -128.03643798828125, y: 84.52842712402344, z: -340.7557373046875
x: -127.98820495605469, y: 84.30717468261719, z: -341.07861328125
x: -128.1005096435547, y: 84.14682006835938, z: -340.7934875488281
```

Рисунок 4.7 – Приклад набору даних магнітометра, виведених в консоль

Оскільки акселерометр, гіроскоп та магнітометр вимірюють значення у трьох координатах (x, y, z), то для генерації наступних бітів послідовності автором пропонується виконувати такі операції:

- конкатенація молодших біт трьох координат;
- отримання молодших біт суми трьох координат;
- отримання молодших біт результату операції XOR трьох координат.

Для кожної з операцій пропонується отримати 1, 2, 4, 8, 16 та 32 біти.

В результаті аналізу отриманих з датчиків гіроскопу та магнітометру даних зроблено висновок, що в цьому випадку доцільно використовувати не молодші біти значень координат, а виконати їх арифметичний зсув на 32 біта вправо, оскільки молодша половина біт у цих датчиків майже завжди складалась виключно з нулів. Приклад двійкового вигляду даних гіроскопа і магнітометра приведено на рис. 4.8 та 4.9.

```
x: -10000001000100110001111011111100100000000000000000000000000000
y: 11111100101100011010011111100101000000000000000000000000000000
z: 11111101101110110001100100011011000000000000000000000000000000
x: -100000010001000110000111110101100000000000000000000000000000000
y: 11111101100101101110010000100110100000000000000000000000000000
z: 11111101100000101101111101001110100000000000000000000000000000
x: -100000010101010110111100101000100100000000000000000000000000000
y: 11111101100010001010110011101110000000000000000000000000000000
z: 11111101100001001110001101011101100000000000000000000000000000
x: 11111101011011001000000111010010100000000000000000000000000000
y: 11111100011110101001011110001000100000000000000000000000000000
z: 11111100100100100110000100101001100000000000000000000000000000
x: -10000001011010000101001011110001100000000000000000000000000000
y: -10000001001110001011010111001110100000000000000000000000000000
z: 11111100110000101100111110101011000000000000000000000000000000
x: 11111101100101100011101001011101100000000000000000000000000000
y: -10000001001101010110001000101000000000000000000000000000000000
z: 11111101001100011001011000000110000000000000000000000000000000
x: -10000001000110010110101001000110010000000000000000000000000000
y: -10000001011100111010111110100011110000000000000000000000000000
z: 11111101001111101111110101111011000000000000000000000000000000
x: -100000010010011011000001111011111100000000000000000000000000000
y: 11111101100010011111110000001000100000000000000000000000000000
z: 11111101011001100100101100110011000000000000000000000000000000
```

Рисунок 4.8 – Двійковий вигляд даних датчика гіроскопу

```
x: -11111110011111110101100010111111100000000000000000000000000000
y: 10000000101010100000110111001010000000000000000000000000000000
z: -11111110001010111001110100100111000000000000000000000000000000
x: -11111110011111110101110101111011000000000000000000000000000000
y: 10000000101010100010000001110001000000000000000000000000000000
z: -11111110001010111001100111111111000000000000000000000000000000
x: -11111110011111110110110001000111100000000000000000000000000000
y: 10000000101010100000110101101100000000000000000000000000000000
z: -11111110001010111000110110010100100000000000000000000000000000
x: -11111110011111110110101001010110100000000000000000000000000000
y: 10000000101010100000011001111010000000000000000000000000000000
z: -11111110001010111001101110110100000000000000000000000000000000
x: -11111110011111110100100110010011100000000000000000000000000000
y: 10000000101010011110111000110100000000000000000000000000000000
z: -11111110001010111010011011000011000000000000000000000000000000
x: -11111110011111110110011110101010100000000000000000000000000000
y: 10000000101010100000100110110001100000000000000000000000000000
z: -11111110001010111001110001010010000000000000000000000000000000
x: -11111110011111110111100101011011100000000000000000000000000000
y: 10000000101010100011100000100001000000000000000000000000000000
z: -11111110001010111010100001101100000000000000000000000000000000
x: -11111110011111110101100011100000000000000000000000000000000000
y: 10000000101010100010011001001110000000000000000000000000000000
z: -11111110001010111010010011101111000000000000000000000000000000
```

Рисунок 4.9 – Двійковий вигляд даних датчика магнітометру

Таким чином, в роботі прийнято рішення провести серію експериментів з отримання випадкових чисел за зазначеними вище принципами та визначити їх ефективність.

Під час отримання кожного значення датчику, всі три координати конвертувались у цілі числа, що представляли собою бітову послідовність координат по стандарту IEEE 754 [28]. Функція, яка була використана для конвертації приведена на рис. 4.10.

```
int doubleToIntBits(double value) {
  ByteData byteData = ByteData(8);
  byteData.setFloat64(0, value, Endian.big);

  return byteData.getInt64(0, Endian.big);
}
```

Рисунок 4.10 – Функція конвертації дійсного числа у його бітове представлення в цілому вигляді

Приклади виконання всіх зазначених вище операцій представлені на рис. 4.11 – 4.13, відповідно.

x: 0.1196012878417968 8
y: 8.937314758300781
z: 3.995341644287109 7

Шістнадцятковий вигляд:

x: 3F4A80487C2B452E
y: 4021dfe7b851eb85
z: 400ff675ae147ae2

конкат. 4 молодших біт: 1110 0101 0010

111001010010

Рисунок 4.11 – Приклад виконання операції конкатенації чотирьох молодших біт координат датчика

```
x: 0.1196012878417968 8
y: 8.937314758300781
z: 3.995341644287109 7

Шістнадцятковий вигляд:
x: 3F4A80487C2B452E
y: 4021dfe7b851eb85
z: 400ff675ae147ae2

    1110
  + 0101
  + 0010
  ------
  ...0101
```

Рисунок 4.12 – Приклад виконання операції додавання чотирьох молодших біт координат датчика

```
x: 0.1196012878417968 8
y: 8.937314758300781
z: 3.995341644287109 7

Шістнадцятковий вигляд:
x: 3F4A80487C2B452E
y: 4021dfe7b851eb85
z: 400ff675ae147ae2

    1110
  ⊕ 0101
  ⊕ 0010
  ------
    1001
```

Рисунок 4.13 – Приклад виконання операції XOR чотирьох молодших біт координат датчика

Для наступних експериментів отримано 10000 значень кожного з датчиків. Усі 10000 значень переведено у двійковий вигляд і записано у відповідний файл. Приклади файлів зі згенерованими послідовностями за допомогою акселерометра, гіроскопа і магнітометра приведено на рис. 4.14 - 4.16 відповідно.

Рисунок 4.14 – Приклад даних акселерометра у двійковому вигляді

Рисунок 4.15 – Приклад даних гіроскопу у двійковому вигляді

Рисунок 4.16 – Приклад даних магнітометру у двійковому вигляді

Усі згенеровані послідовності перевірено набором тестів NIST [7]. Результати тестів представлені у табл. 4.1-4.3 для кожного з датчиків.

Операція конкат. x,y,z позначає конкатенацію, операція x+y+z позначає суму, а x^y^z позначає операцію XOR зазначеної кількості біт.

Зелений колір позначає успішне проходження тесту, червоний – тест не пройдено, сірий – згенерована послідовність не відповідає вимогам тесту по довжині.

Таблиця 4.1 – Результати тестів NIST для згенерованих послідовностей за допомогою акселерометра

Операція	Кількість біт	Тест №1	Тест №2	Тест №3	Тест №4	Тест №5	Тест №6	Тест №7	Тест №8	Тест №9	Тест №10	Тест №11	Тест №12	Тест №13	Тест №14	Тест №15
конкат. x,y,z	1	+	+	+	+	+	+	+				+	-	+		
конкат. x,y,z	2	+	+	+	+	+	+	+				+	-	+		
конкат. x,y,z	4	+	+	+	+	+	+	-				+	-	+		
конкат. x,y,z	8	+	+	+	-	+	+	-				-	-	+		
конкат. x,y,z	16	-	+	-	-	-	-	+				-	-	-		
конкат. x,y,z	32	-	-	+	-	-	-	+				-	-	-		
x+y+z	1	-	-	-	-	+	-	-				-	-	-		
x+y+z	2	-	-	-	-	+	-	-				-	-	-		
x+y+z	4	-	-	-	-	+	+	+				+	-	-		
x+y+z	8	-	+	-	-	+	+	-				-	-	-		
x+y+z	16	-	-	-	-	-	-	+				-	-	-		
x+y+z	32	-	-	-	-	-	-	+				-	-	-		
x^y^z	1	+	+	+	+	+	+	+				+	+	+		
x^y^z	2	+	+	+	+	+	+	+				+	+	+		
x^y^z	4	+	+	+	+	+	+	+				+	+	+		
x^y^z	8	+	+	+	+	+	+	+				+	-	+		
x^y^z	16	+	+	+	+	+	+	+				-	-	+		
x^y^z	32	+	+	+	-	-	-	+				-	-	+		

Таблиця 4.2 – Результати тестів NIST для згенерованих послідовностей за допомогою гіроскопа

Операція	Кількість біт	Тест №1	Тест №2	Тест №3	Тест №4	Тест №5	Тест №6	Тест №7	Тест №8	Тест №9	Тест №10	Тест №11	Тест №12	Тест №13	Тест №14	Тест №15
конкат. x,y,z	1	+	+	+	+	+	+	+				+	+	+		
конкат. x,y,z	2	+	+	+	+	+	+	+				+	+	+		
конкат. x,y,z	4	+	+	+	+	+	+	+				+	+	+		
конкат. x,y,z	8	+	+	+	+	+	+	+				+	+	+		
конкат. x,y,z	16	+	+	+	+	+	+	+				+	-	+		
конкат. x,y,z	32	-	-	-	-	-	-	+				-	-	-		
x+y+z	1	+	+	+	+	+	+	+				+	+	+		
x+y+z	2	+	+	+	+	+	+	+				+	+	+		
x+y+z	4	+	+	+	+	+	+	+				+	-	+		
x+y+z	8	+	+	+	+	+	+	+				+	+	+		
x+y+z	16	+	+	+	+	+	+	+				+	-	+		
x+y+z	32	-	-	-	-	-	-	+				-	-	-		
x^y^z	1	+	+	+	+	+	+	+				+	+	+		
x^y^z	2	+	+	+	+	+	+	+				+	-	+		
x^y^z	4	+	+	+	+	+	+	+				+	+	+		
x^y^z	8	+	+	+	+	+	+	+				+	-	+		
x^y^z	16	+	+	+	+	+	+	+				+	+	+		
x^y^z	32	-	-	-	-	-	-	+				-	-	-		

Таблиця 4.3 – Результати тестів NIST для згенерованих послідовностей за допомогою магнітометра

Операція	Кількість біт	Т№1	Тест №2	Тест №3	Тест №4	Тест №5	Тест №6	Тест №7	Тест №8	Тест №9	Тест №10	Тест №11	Тест №12	Тест №13	Тест №14	Тест №15
конкат. x,y,z	1	+	+	+	+	+	+	+				+	-	+		
конкат. x,y,z	2	+	+	+	+	+	+	+				+	+	+		
конкат. x,y,z	4	+	+	+	+	+	+	+				+	+	+		
конкат. x,y,z	8	+	+	+	+	+	+	-				+	-	+		
конкат. x,y,z	16	-	-	-	-	+	-	+				+	-	-		
конкат. x,y,z	32	-	-	-	-	-	-	+				-	-	-		
x+y+z	1	+	+	+	+	+	+	+				+	+	+		
x+y+z	2	+	+	+	+	+	+	+				+	+	+		
x+y+z	4	+	+	+	+	+	+	+				+	+	+		
x+y+z	8	+	+	+	+	+	+	+				+	+	+		
x+y+z	16	-	-	-	+	-	-	+				-	-	-		
x+y+z	32	-	-	-	-	-	-	+				-	-	-		
x^y^z	1	+	+	+	+	+	+	+				+	-	+		
x^y^z	2	+	+	+	+	+	+	+				+	-	+		
x^y^z	4	+	+	+	+	+	+	+				+	-	+		
x^y^z	8	+	+	+	+	+	+	+				+	+	+		
x^y^z	16	-	-	-	-	+	+	+	+		+	+	-	-		
x^y^z	32	-	-	-	+	-	-	+	+		+	-	-	-		

Тести пронумеровано за їх порядком у кейсі NIST:

- тест №1 – Frequency (Monobit) Test;
- тест №2 – Frequency Test within a Block;
- тест №3 – Runs Test;
- тест №4 – Test for the Longest Run of Ones in a Block;
- тест №5 – Binary Matrix Rank Test;
- тест №6 – Discrete Fourier Transform (Spectral) Test;
- тест №7 – Non-overlapping Template Matching Test;
- тест №8 – Overlapping Template Matching Test;
- тест №9 – Maurer's "Universal Statistical" Test;
- тест №10 – Linear Complexity Test;
- тест №11 – Serial Test;
- тест №12 – Approximate Entropy Test;
- тест №13 – Cumulative Sums (Cusum) Test;
- тест №14 – Random Excursions Test;
- тест №15 – Random Excursions Variant Test.

Як видно з таблиць, найбільш ефективними способами генерації виявились:

- для акселерометра – операція XOR для 4 молодших біт;
- для гіроскопа – операція XOR для 16 молодших біт;
- для магнітометра – операція додавання та XOR для 8 молодших біт.

Для реалізації генерації випадкових чисел у розроблюваному комплексі, використовуватиметься операція XOR 4 молодших біт для акселерометра, операція XOR 16 молодших біт для гіроскопа та операція XOR 8 молодших біт для магнітометра.

4.3 Оцінка якості випадкових чисел, згенерованих за допомогою розробленого комплексу

Для перевірки якості чисел згенерованих за допомогою розроблено комплексу було згенеровано 2000 чисел по модулю 1024 для таких варіантів режимів:

- генерація випадкових чисел за допомогою акселерометра;

- генерація випадкових чисел за допомогою гіроскопа;
- генерація випадкових чисел за допомогою магнітометра;
- генерація псевдовипадкових чисел згенерованих алгоритмом BBS із зерном згенерованим за допомогою акселерометра;
- генерація псевдовипадкових чисел згенерованих алгоритмом BBS із зерном згенерованим за допомогою гіроскопа;
- генерація псевдовипадкових чисел згенерованих алгоритмом BBS із зерном згенерованим за допомогою магнітометра.

Приклади декількох початкових рядків файлів зі згенерованими випадковими числами за допомогою акселерометра у десятковому та двійковому вигляді приведені на рис. 4.17 та 4.18, а їх повний вміст – у додатках К відповідно.

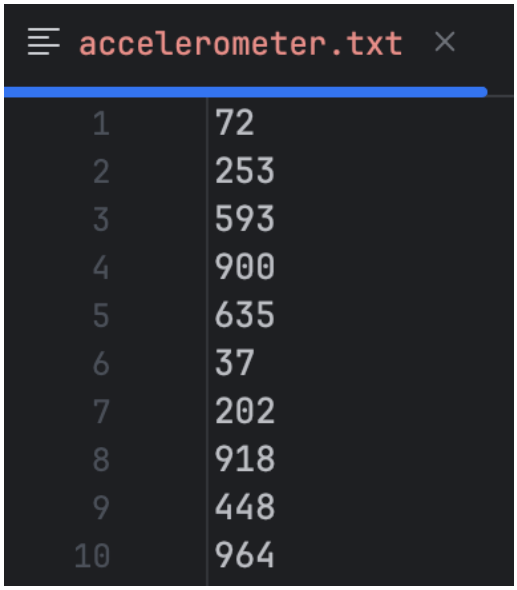

Рисунок 4.17 – Початкові рядки файлу зі згенерованими випадковими числами за допомогою акселерометра

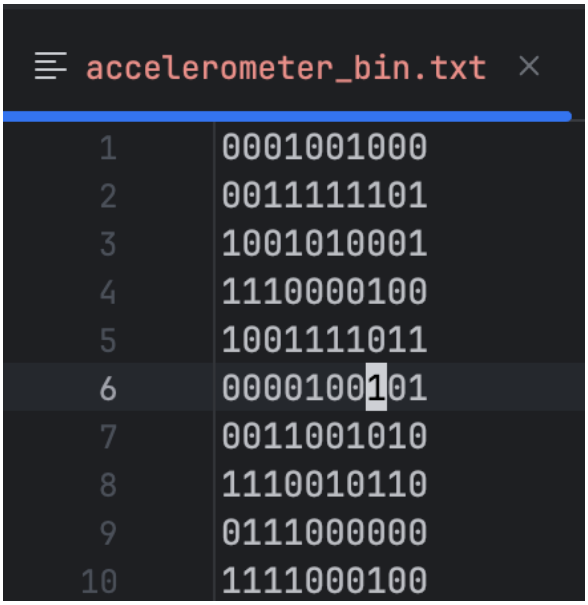

Рисунок 4.18 – Початкові рядки файлу із двійковим представленням випадкових чисел згенерованих за допомогою акселерометра

Приклади декількох початкових рядків файлів зі згенерованими випадковими числами за допомогою гіроскопа у десятковому та двійковому вигляді приведені на рис. 4.19 та 4.20, а їх повний вміст – у додатку Л.

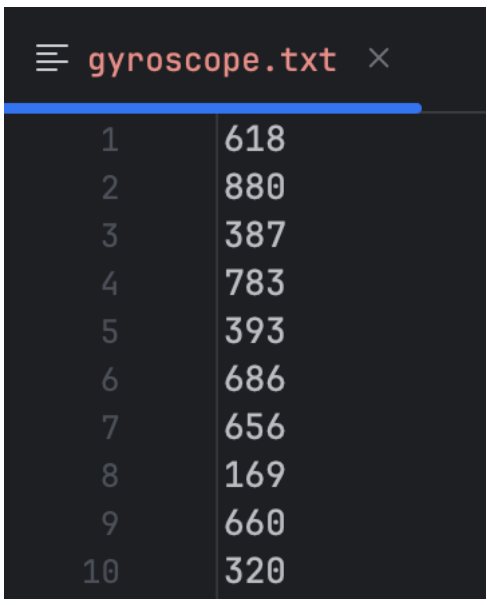

Рисунок 4.19 – Початкові рядки файлу зі згенерованими випадковими числами за допомогою гіроскопа

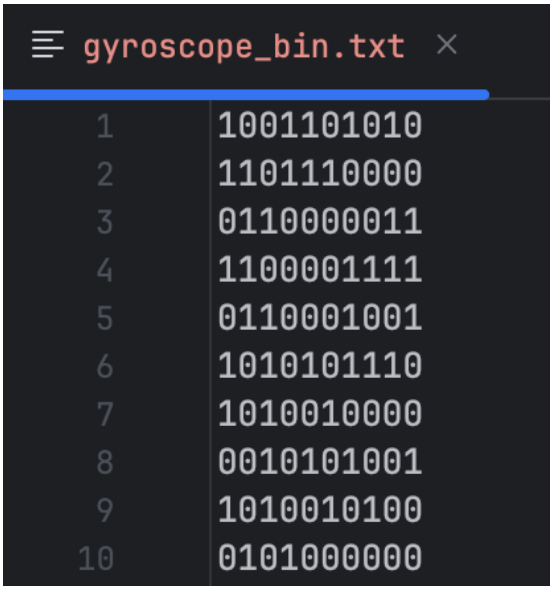

Рисунок 4.20 – Початкові рядки файлу із двійковим представленням випадкових чисел згенерованих за допомогою гіроскопа

Приклади декількох початкових рядків файлів зі згенерованими випадковими числами за допомогою магнітометра у десятковому та двійковому вигляді приведені на рис. 4.21 та 4.22, а їх повний вміст – у додатку М.

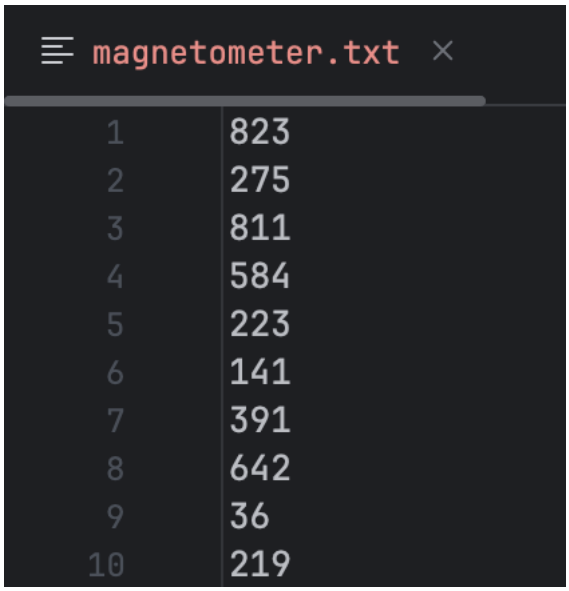

Рисунок 4.21 – Початкові рядки файлу зі згенерованими випадковими числами за допомогою магнітометра

magnetometer_bin.txt

```
1100110111
0100010011
1100101011
1001001000
0011011111
0010001101
0110000111
1010000010
0000100100
0011011011
```

Рисунок 4.22– Початкові рядки файлу із двійковим представленням випадкових чисел згенерованих за допомогою магнітометра

Приклади декількох початкових рядків файлів зі згенерованими псевдовипадковими числами згенерованими алгоритмом BBS із зерном згенерованим за допомогою акселерометра у десятковому та двійковому вигляді приведені на рис. 4.23 та 4.24, а їх повний вміст – у додатку Н.

bbs_accelerometer.txt

```
181
116
287
330
464
466
617
741
496
274
```

Рисунок 4.23 – Початкові рядки файлу зі згенерованими псевдовипадковими числами за допомогою алгоритму BBS із зерном згенерованим за допомогою акселерометра

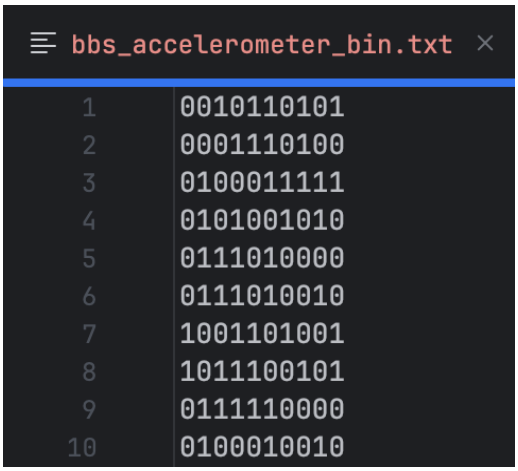

Рисунок 4.24 – Початкові рядки файлу із двійковим представленням псевдовипадкових чисел згенерованих за допомогою алгоритму BBS із зерном згенерованим за допомогою акселерометра

Приклади декількох початкових рядків файлів зі згенерованими псевдовипадковими числами згенерованими алгоритмом BBS із зерном згенерованим за допомогою гіроскопа у десятковому та двійковому вигляді приведені на рис. 4.25 та 4.26, а їх повний вміст – у додатку П.

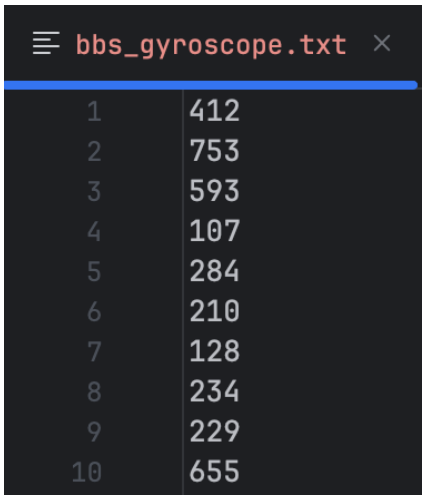

Рисунок 4.25 – Початкові рядки файлу зі згенерованими псевдовипадковими числами за допомогою алгоритму BBS із зерном згенерованим за допомогою гіроскопа

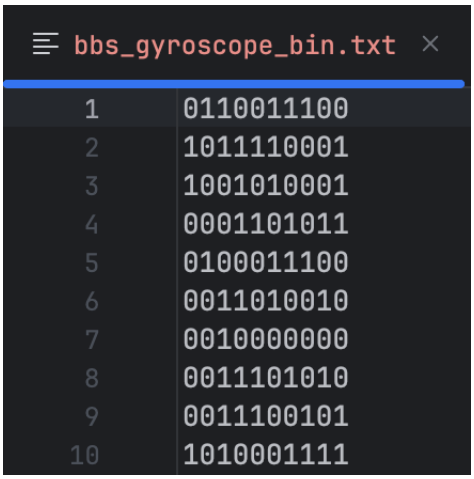

Рисунок 4.26 – Початкові рядки файлу із двійковим представленням псевдовипадкових чисел згенерованих за допомогою алгоритму BBS із зерном згенерованим за допомогою гіроскопа

Приклади декількох початкових рядків файлів зі згенерованими псевдовипадковими числами згенерованими алгоритмом BBS із зерном згенерованим за допомогою магнітометра у десятковому та двійковому вигляді приведені на рис. 4.27 та 4.28, а їх повний вміст – у додатку Р.

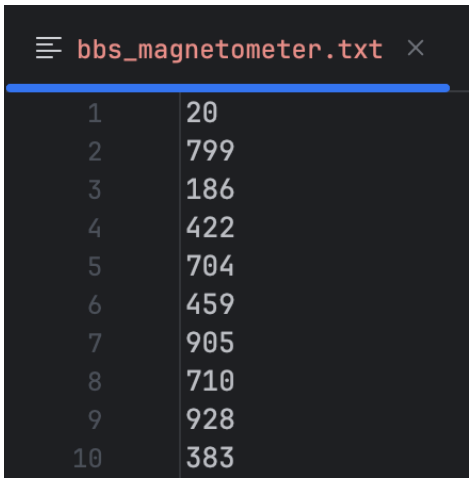

Рисунок 4.27 – Початкові рядки файлу зі згенерованими псевдовипадковими числами за допомогою алгоритму BBS із зерном згенерованим за допомогою магнітометра

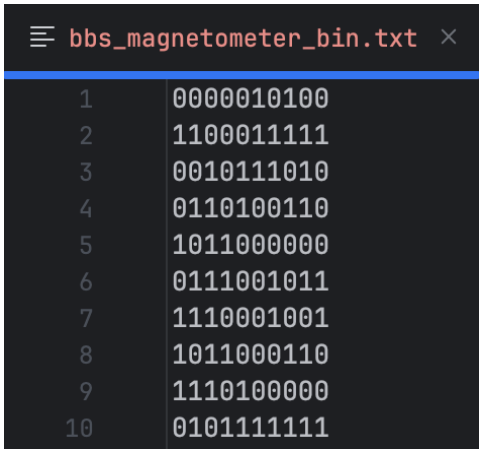

Рисунок 4.28 – Початкові рядки файлу із двійковим представленням псевдовипадкових чисел згенерованих за допомогою алгоритму BBS із зерном згенерованим за допомогою магнітометра

Таблиці результатів тестування згенерованих послідовностей чисел за допомогою утиліти NIST [27] представлені у табл. 4.4.

На рис. 4.29 – 4.34 представлені результати графічних тестів для кожного з режимів генерації.

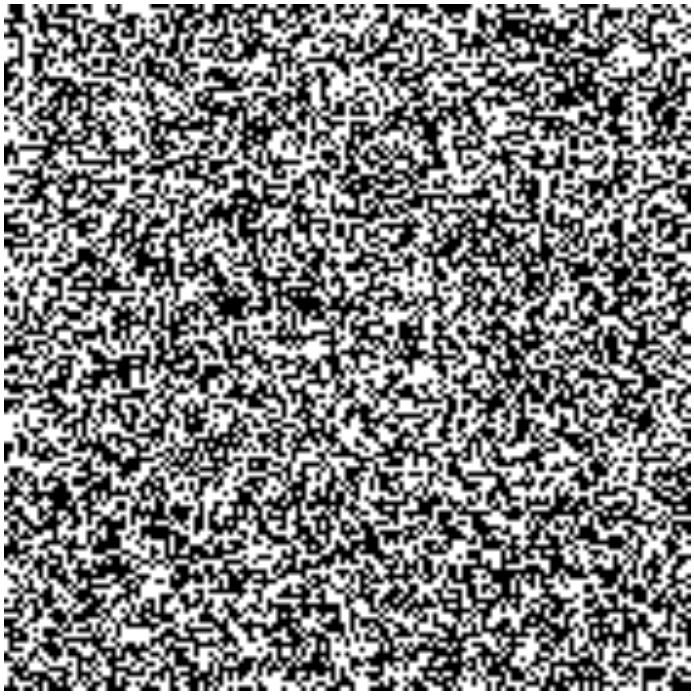

Рисунок 4.29 – Результат графічного тесту для випадкових чисел згенерованих за допомогою акселерометра

Таблиця 4.4 – Результати тестування згенерованих послідовностей чисел

Назва тесту	Акселерометр	Гіроскоп	Магнітометр	BBS+зерно акселерометра	BBS+зерно гіроскопа	BBS+зерно магнітометра
Frequency (Monobit) Test	+	+	+	+	+	+
Frequency Test within a Block	+	+	+	+	+	+
Runs Test	+	+	+	+	+	+
Test for the Longest Run of Ones in a Block	+	+	+	+	+	+
Binary Matrix Rank Test	+	+	+	+	+	+
Discrete Fourier Transform (Spectral) Test	+	+	+	+	+	+
Non-overlapping Template Matching Test	+	+	+	+	+	+
Overlapping Template Matching Test						
Maurer’s “Universal Statistical” Test						
Linear Complexity Test						
Serial Test	+	+	+	+	+	+
Approximate Entropy Test	+	+	+	+	+	+
Cumulative Sums (Cusum) Test	+	+	+	+	+	+
Random Excursions Test						
Random Excursions Variant Test						

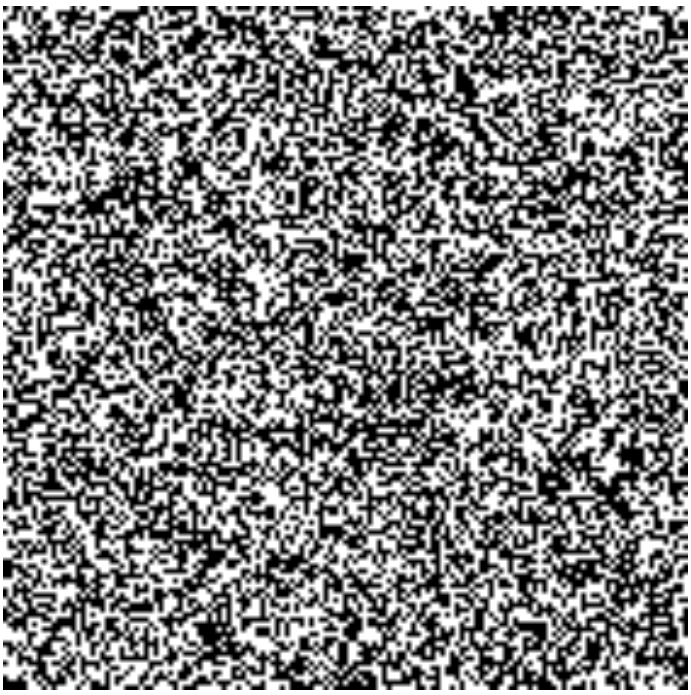

Рисунок 4.30 – Результат графічного тесту для випадкових чисел згенерованих за допомогою гіроскопа

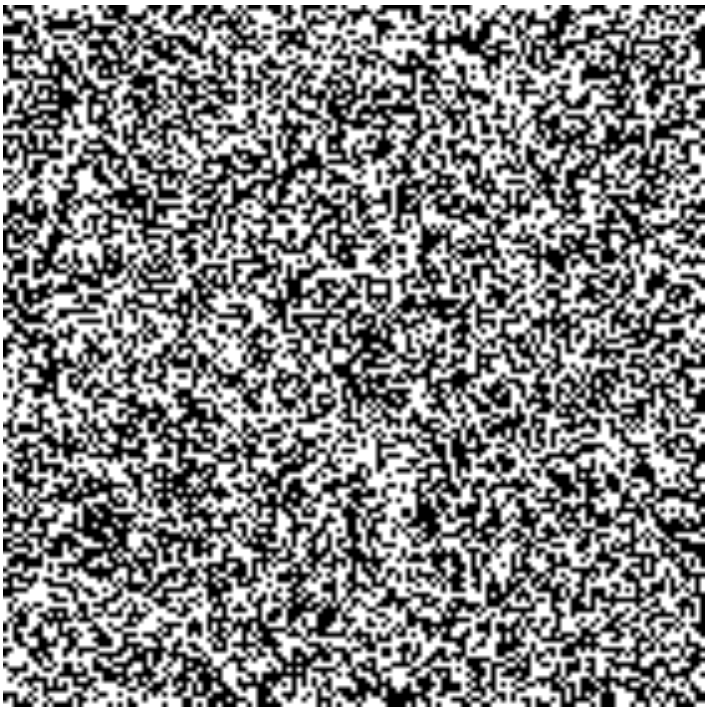

Рисунок 4.31 – Результат графічного тесту для випадкових чисел згенерованих за допомогою магнітометра

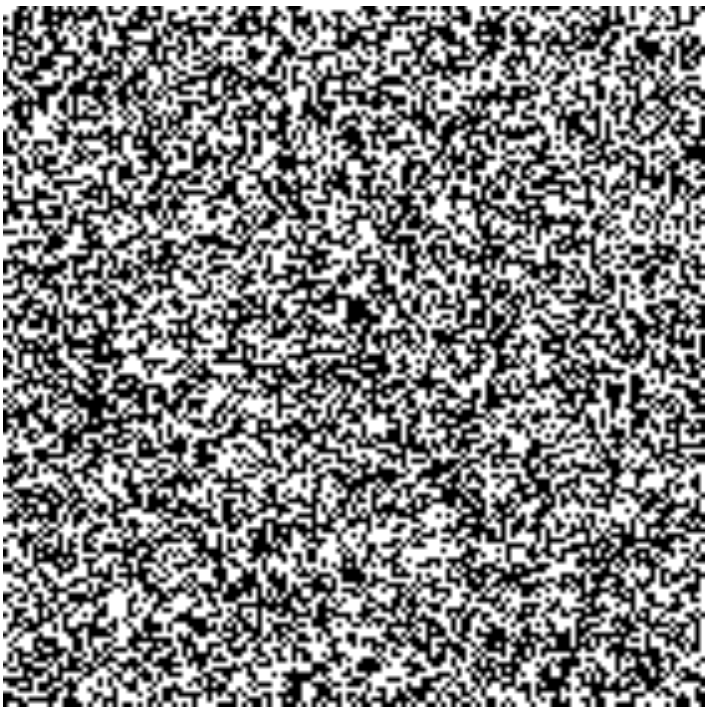

Рисунок 4.32 – Результат графічного тесту для псевдовипадкових чисел згенерованих за допомогою алгоритму BBS із зерном згенерованим за допомогою акселерометра

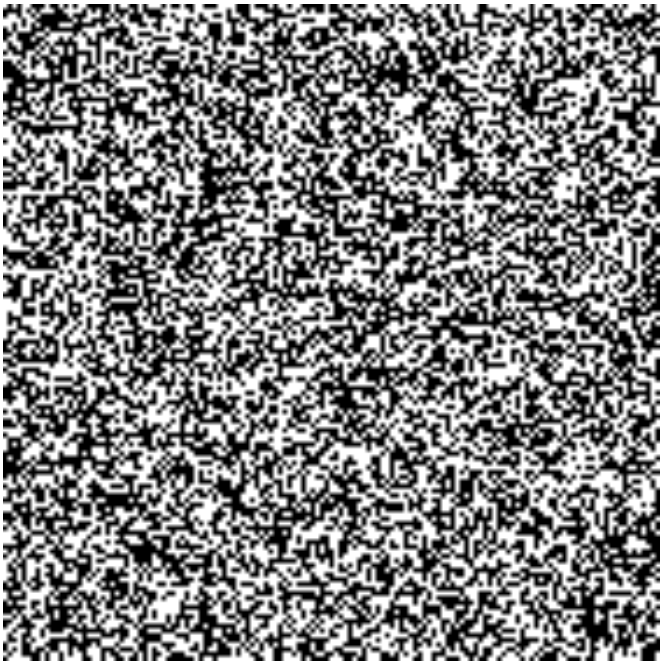

Рисунок 4.33 – Результат графічного тесту для псевдовипадкових чисел згенерованих за допомогою алгоритму BBS із зерном згенерованим за допомогою гіроскопа

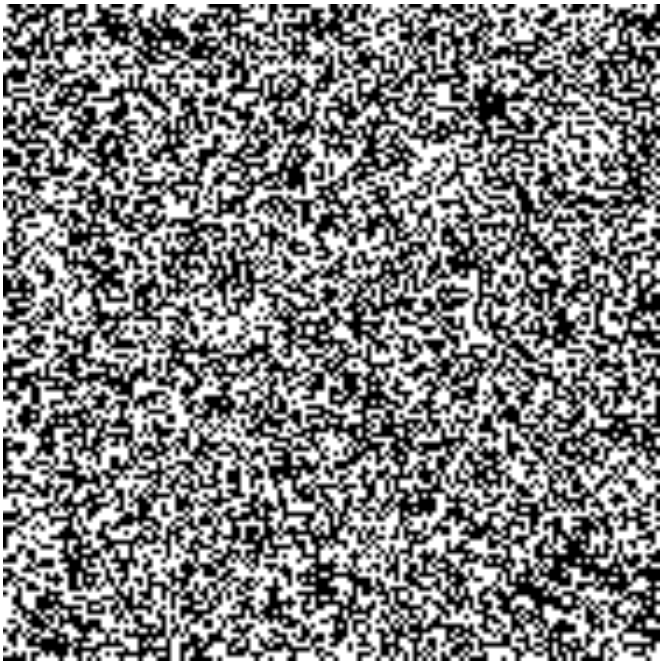

Рисунок 4.34 – Результат графічного тесту для псевдовипадкових чисел згенерованих за допомогою алгоритму BBS із зерном згенерованим за допомогою магнітометра

Швидкість генерації чисел для кожного режиму представлена у табл. 4.5.

Таблиця 4.5 – Швидкість генерації чисел для кожного режиму

Режим	Орієнтовний час генерації 1000 чисел
Акселерометр	27хв 10с
Гіроскоп	7хв 9с
Магнітометр	13хв 32с
BBS	66.2мс

Результати тестів показали, що коли необхідна висока швидкодія, генератор псевдовипадкових чисел BBS із випадковим зерном, дає схожі за якістю результати із генераторами випадкових чисел на базі датчиків мобільного пристрою.

4.4 Порівняння характеристик випадкових чисел, згенерованих за допомогою генераторів різних типів

Результати перевірки згенерованих чисел на випадковість за допомогою мобільного пристрою в даній роботі можна порівняти з результати отриманими в роботі [29]. У вказаній роботі отримані характеристики ступеню випадковості чисел, згенерованих за допомогою пристрою, що використовує шуми напівпровідникового приладу (стабілітрону).

Виходячи з отриманих результатів, генерація випадкових чисел за допомогою датчиків, що доступні в мобільних пристроях дає гарні результати, як і генерація випадкових чисел на базі пристрою, що використовує шуми стабілітрона.

4.5 Висновки за розділом

Для перевірки якості згенерованих випадкових послідовностей обрано кейс тестів NIST та візуальний графічний тест. Проведено експерименти для перевірки різних способів генерації випадкових послідовностей за допомогою використаних датчиків, на основі яких визначено ефективні способи для кожного датчика. Оцінено якість отриманих випадкових чисел в можливих режимах їх генерування.

ВИСНОВКИ ТА РЕКОМЕНДАЦІЇ

У роботі розроблено апаратно-програмний комплекс генерації випадкових чисел на базі мобільного пристрою та проведено дослідження його ефективності.

Розглянуто основні поняття в сфері генерації випадкових чисел та основні джерела ентропії, що доступні в мобільному пристрої, проведено їх порівняльний аналіз.

Розроблено архітектуру комплексу та систему команд для обміну даними між всіма його елементами. Для створення пристрою генерації випадкових чисел використано мобільний пристрій IPhone 14 та джерела ентропії – датчики акселерометра, гіроскопа та магнітометра.

Обрані середовища, мови та засоби розробки для клієнтської, серверної та мобільної частин комплексу. Розроблено основні алгоритми та програмне забезпечення, виконано перевірку його працездатності.

Виконане експериментальне дослідження якості отриманих випадкових чисел статистичними та візуальним тестом.

Розроблений комплекс може використовуватися на практиці для отримання випадкових та псевдовипадкових чисел та у навчальних цілях.

ПЕРЕЛІК ПОСИЛАНЬ

1. Опрятний А.О., Остапець Д.О. Генерація випадкових чисел на базі мобільних пристроїв. *Сучасні інформаційні і комунікаційні технології на транспорті, в промисловості та освіті*: Тези XV Міжнародної науково-практичної конференції, м. Дніпро, 16-17 груд. 2021 р. Дніпро: ДІІТ, 2021. С. 199.
2. Опрятний А.О., Остапець Д.О. Генерація випадкових чисел на базі мобільних пристроїв. *Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості і освіті*: Тези XVI Міжнародної науково-практичної конференції, м. Дніпро, 14-15 груд. 2022 р. Дніпро: ДІІТ, 2022. С. 150.
3. Опрятний А.О., Остапець Д.О. Використання датчиків смартфона для генерації випадкових чисел. *Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості і освіті*: Тези XVIII Міжнародної науково-практичної конференції, м. Дніпро, 12-13 груд. 2024р. Дніпро: УДУНТ, 2024. С. 184.
4. Опрятний А.О. Генерація випадкових чисел за допомогою смартфонів. *Наука і сталий розвиток транспорту 2024. Т. II*: зб. тез доп. Всеукр. наук.-техн. конф. студентів і молодих учених, Дніпро, 27 листоп. 2024 р. Дніпро: УДУНТ, 2024. С. 21.
5. Подорожняк А. О., Токарев М. Г. Метод генерації псевдовипадкових чисел високої стійкості. *Вісник Національного технічного університету "ХПІ"*. 2017. № 50 (1271).
6. Генератори випадкових чисел. *Вікі ЦДУ*. URL: https://wiki.cuspu.edu.ua/index.php/Генератори_випадкових_чисел (дата звернення: 25.10.2024).
7. National Institute of Standards and Technology Special Publication 800-22 revision 1a Natl. Inst. Stand. Technol. Spec. Publ. 800-22rev1a, 131 pages (April 2010).
8. Sensors Overview. *Android Developers*. URL: https://developer.android.com/develop/sensors-and-location/sensors/sensors_overview (дата звернення: 07.01.2025).

9. Introduction to Randomness and Random Numbers. *Random.org*. URL: https://www.random.org/randomness/ (дата звернення: 26.10.2024).
10. Motion sensors. *Android Developers*. URL: https://developer.android.com/develop/sensors-and-location/sensors/sensors_motion (дата звернення: 07.01.2025).
11. Hennebert, C., Hossayni H., Lauradoux, C. Entropy harvesting from physical sensors. *Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks*, м. Будапешт, 17–19 квіт. 2013 р. Будапешт, 2013. С. 149–154.
12. Voris, J., Saxena N., Halevi T. Accelerometers and Randomness: Perfect Together. *Fourth ACM Conference on Wireless Network Security*, м. Гамбург, 11–14 черв. 2011 р. Гамбург, 2011. С. 115–126.
13. Position sensors. *Android Developers*. URL: https://developer.android.com/develop/sensors-and-location/sensors/sensors_position (дата звернення: 07.01.2025).
14. Suciu A., Lebu D., Marton K. Unpredictable Random Number Generator Based on Mobile Sensors. *IEEE International Conference on Intelligent Computer Communication and Processing*, м. Клуж-Напока, 25-27 серп. 2011 р. Клуж-Напока. 2011.
15. Environment sensors. *Android Developers*. URL: https://developer.android.com/develop/sensors-and-location/sensors/sensors_environment (дата звернення: 07.01.2025).
16. Pawlowski, M.P., Jara, A., Ogorzalek, M., Jara A. J. Harvesting Entropy for Random Number Generation for Internet of Things Constrained Devices Using On-Board Sensors. 2015.
17. Cho, S.M., Hong, E., Seo, S.H. Random Number Generator Using Sensors for Drone. *IEEE Access*. 2020.
18. Tanenbaum A. S., Wetherall D. J., Computer networks – 5th ed. 1944. 552 с.
19. Blum L., Blum M., Shub M., A Simple Unpredictable Pseudo-Random Number Generator. *SIAM Journal on Computing*. 1986. С. 364-383.

20. Build simple, secure, scalable systems with Go. *Go*. URL: https://go.dev (дата звернення 14.01.2025).
21. Goland – Go Productive. *JetBrains*. URL: https://www.jetbrains.com/go/ (дата звернення 14.01.2025).
22. Paint your UI to life. *Dart*. URL: https://dart.dev (дата звернення 14.01.2025).
23. Build for any screen. *Flutter*. URL: https://flutter.dev (дата звернення 14.01.2025).
24. Android Studio. *Android Developers*. URL: https://developer.android.com/studio (дата звернення 16.01.2025).
25. Sensors_plus. *Pub.dev*. URL: https://pub.dev/packages/sensors_plus (дата звернення 16.01.2025).
26. Rseudo-Random vs. True Random. *Bo Allen*. URL: https://boallen.com/random-numbers.html (дата звернення 22.01.2025).
27. NIST Randomness Testsuite. *GitHub*. URL: https://github.com/stevenang/randomness_testsuite (дата звернення 20.01.2025).
28. IEEE Standard for Binary Floating-Point Arithmetic (IEEE Std 754-1985). *An Americal National Standard.* 2008.
29. Маслак. А. В., Дослідження та розробка апартно-програмних комплексів засобів генерації випадкових чисел. Комплекс генерації випадкових чисел на базі мікроконтролерів. Дніпро, 2022. 85с.

ДОДАТОК А

Лістинг програми для мобільного пристрою

ДОДАТОК Б

Лістинг програми функції main для сервера

ДОДАТОК В

Лістинг програми функції main для клієнта

ДОДАТОК Г

Лістинг логіки встановлення режиму і генерації випадкових чисел на сервері

ДОДАТОК Д

Лістинг програми взаємодії сервера з мобільним пристроєм

ДОДАТОК Ж

Лістинг програми реалізації алгоритма BBS

ДОДАТОК И

Лістинг структури конфігурації серверу

```
}
```

ДОДАТОК К

Згенеровані випадкові числа за допомогою акселерометра

ДОДАТОК Л

Згенеровані випадкові числа за допомогою гіроскопа

ДОДАТОК М

Згенеровані випадкові числа за допомогою магнітометра

ДОДАТОК Н

Згенеровані псевдовипадкові числа за допомогою алгоритма BBS із зерном, що згенероване за допомогою акселерометра

ДОДАТОК П

Згенеровані псевдовипадкові числа за допомогою алгоритма BBS із зерном, що згенероване за допомогою гіроскопа

ДОДАТОК Р

Згенеровані псевдовипадкові числа за допомогою алгоритма BBS із зерном, що згенероване за допомогою магнітометра