

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**приладобудівний факультет  
кафедра автоматизації та систем неруйнівного контролю**

До захисту допущено:

В.о. завідувача кафедри

\_\_\_\_\_Юрій КИРИЧУК

«\_\_»\_\_\_\_\_20\_\_р.

**Магістерська дисертація**

**на здобуття ступеня магістра**

**зі спеціальності 151 «Автоматизація та комп'ютерно-інтегровані  
технології»**

**на тему: «Автоматизована система вимірювання напруження  
електричного поля на місцевості»**

Виконав:

студент VI курсу, групи ПК-01мп Белак Артур Сергійович

\_\_\_\_\_

Керівник:

кандидат технічних наук, доцент,

Баженов Віктор Григорович

\_\_\_\_\_

Консультант з розроблення стартап-проекту:

доктор економічних наук, професор,

Бояринова К.О.

\_\_\_\_\_

Рецензент:

доктор технічних наук, професор,

Єременко Володимир Станіславович

\_\_\_\_\_

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

Студент \_\_\_\_\_

Київ – 2021 року

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**  
**Приладобудівний факультет**  
**Кафедра автоматизації та систем неруйнівного контролю**

Рівень вищої освіти – другий (магістерський)

Спеціальність – 151 «Автоматизація та комп'ютерно-інтегровані технології»

Освітньо-професійна програма «Комп'ютерно-інтегровані системи та технології неруйнівного контролю і діагностики»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

\_\_\_\_\_ Юрій КИРИЧУК

«\_\_» \_\_\_\_\_ 20\_\_р.

**ЗАВДАННЯ**  
**на магістерську дисертацію студенту**  
**Белаку Артуру Сергійовичу**

1. Тема дисертації «Автоматизована система вимірювання напруження електричного поля на місцевості», науковий керівник дисертації Баженов Віктор Григорович, кандидат технічних наук, доцент, затверджена наказом по університету від «\_\_» \_\_\_\_\_ 20\_\_р. № \_\_\_\_\_

2. Термін подання студентом дисертації

---

3. Об'єкт дослідження: процес вимірювання напруженості електростатичного поля

4. Вихідні дані: Об'єкт дослідження - процес вимірювання напруженості електростатичного поля

5. Перелік завдань, які потрібно розробити: оглянути методи вимірювання напруженості електростатичного поля атмосфери, проаналізувати та обрати метод для вимірювання напруженості, оглянути методи автоматизації вимірювань, обрати методи автоматизації, розробити автоматизовану систему, оглянути методи передачі даних, розробити програму для відображення результатів.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: 3 плакати

7. Орієнтовний перелік публікацій: учасник «Sikorsky Challenge 2021: Україна і світ»

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розробка стартап-проектів	Бояринова К.О., д.е.н., професор; професор кафедри менеджменту факультету менеджменту та маркетингу КПІ ім. Ігоря Сікорського		

9. Дата видачі завдання \_\_\_\_\_

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Формулювання завдання магістерської дисертації	1 тиждень	
2	Огляд методів вимірювання напруженості електричного поля атмосфери	2 тижні	
3	Розробка вимірювача напруженості електричного поля атмосфери	1 тиждень	
4	Огляд методів автоматизації процесу вимірювання напруженості електричного поля атмосфери на місцевості	2 тиждень	
5	Розробка системи автоматизації для процесу вимірювання електричного поля атмосфери на місцевості	2 тиждень	
6	Огляд бібліотек та можливостей мови програмування C# для створення програм, для відображення результатів вимірів автоматизовано системи для вимірювання напруженості електричного поля атмосфери на місцевості	2 тижні	
7	Розробка програмного забезпечення для відображення результатів вимірів	2 тижні	
8	Розробка стартап-проекту	1 тиждень	
9	Формування висновків та оформлення пояснювальної записки та презентації	1 тиждень	

Студент

Артур БЕЛАК

Науковий керівник

Віктор БАЖЕНОВ

# РЕФЕРАТ

## **Актуальність теми**

На даний момент електричне поле атмосфери є характеристика, яка слабо вивчена. А даремно так як ця характеристика може бути дуже корисною. На даний момент вже є роботи, які описують що електричне поле атмосфери, може слугувати інформативним параметром при знаходженні корисних копалин. Головною характеристикою електричного поля є його напруженість. На даний момент процес вимірювання напруженості електричного поля атмосфери достатньо складний процес, так як в даному разі цей параметр має дуже маленькі показники, які важко фіксувати. І на даний час не існує ні одної автоматизованої системи по вимірювання напруженості електричного поля атмосфери на місцевості, для дослідження корисних копалин, і в загальному впливу різних факторів на напруженості.

Саме тому питання створення автоматизованої системи для вимірювання напруженості електричного поля атмосфери на місцевості, а також покращення існуючих методів такого вимірювання є дуже актуальною.

## **Мета і задачі дослідження**

**Мета дослідження** – покращення методів вимірювання напруженості електричного поля атмосфери, а також розробка методів автоматизації даного процесу.

Для досягнення поставленої мети були розв'язані такі **задачі**:

- 1) Аналіз всіх існуючих методів вимірювання напруженості електричного поля атмосфери.
- 2) Удосконалення динамічного методу вимірювання напруженості електричного поля атмосфери.
- 3) Аналіз способів автоматизації процесу вимірювання напруженості електричного поля атмосфери на місцевості.
- 4) Розробка автоматизованої системи вимірювання напруженості електричного поля атмосфери на місцевості.

**Об'єкт дослідження** – процес автоматизованого вимірювання напруженості електричного поля атмосфери на місцевості.

**Предмет дослідження** – методи вимірювання напруженості електричного поля атмосфери на місцевості.

**Методи дослідження** ґрунтуються на теоретичних основах вимірювань напруженості електричного поля атмосфери, автоматизації процесу, теоретичній основі роботи GPS, а також теоретичних основах комп'ютерної обробки інформації

#### **Наукова новизна одержаних результатів**

- 1) Удосконалено динамічний метод вимірювання напруженості електричного поля атмосфери.

#### **Практичне значення результатів дисертації**

- 1) Розроблено автоматизований прилад для вимірювання напруженості електричного поля атмосфери на місцевості.

#### **Ключові слова**

Вимірювання напруженості електричного поля атмосфери, динамічний метод вимірювання напруженості електричного поля атмосфери, автоматизація процесу.

# ABSTRACT

## **Actuality of theme**

At present, the electric field of the atmosphere is a characteristic that is poorly understood. And in vain as this feature can be very useful. At present, there are works that describe that the electric field of the atmosphere can serve as an informative parameter in finding minerals. The main characteristic of the electric field is its intensity. At the moment, the process of measuring the electric field strength of the atmosphere is quite a complex process, as in this case this parameter has very small values that are difficult to fix. And currently there is no automated system for measuring the electric field strength of the atmosphere in the field, for the study of minerals, and in general the influence of various factors on the voltage.

That is why the issue of creating an automated system for measuring the electric field strength of the atmosphere in the field, as well as improving existing methods of such measurement is very important.

## **The purpose and objectives of the study**

**The purpose of the study** is to improve methods for measuring the electric field strength of the atmosphere, as well as to develop methods for automating this process.

To achieve this goal, the following tasks are **developed**:

- 1) Analysis of all existing methods for measuring the electric field strength of the atmosphere.
- 2) Improving the dynamic method of measuring the electric field strength of the atmosphere.
- 3) Analysis of methods for automating the process of measuring the electric field strength of the atmosphere in the field.
- 4) Development of an automated system for measuring the electric field strength of the atmosphere in the field.

**Object of research** - is the process of automated measurement of the electric field strength of the atmosphere in the field.

**Subject of research** - methods of measuring the electric field strength of the atmosphere in the field.

**Methods of research** are based on the theoretical foundations of measuring the electric field strength of the atmosphere, process automation, the theoretical basis of GPS, as well as the theoretical foundations of computer information processing

**Scientific novelty of research:**

1) The dynamic method of measuring the electric field strength of the atmosphere has been improved.

**Practical value dissertation results:**

1) Developed an automated device for measuring the electric field strength of the atmosphere in the field.

**Keywords**

Atmospheric electric field strength measurement, dynamic method of atmospheric electric field strength measurement, process automation.

## Зміст

<b>Перелік основних скорочень.....</b>	<b>10</b>
<b>ВСТУП.....</b>	<b>11</b>
1. Аналітичний огляд .....	13
1.1. Електричне поле в атмосфері.....	13
1.2. Динамічний метод вимірювання напруженості електричного поля в атмосфері.....	15
1.3. Схема та принцип роботи датчику .....	29
1.4. Висновки до розділу .....	30
2. Розробка автоматизованої системи .....	32
2.1. Технологія GPS .....	32
2.2. Технологія SD-Card.....	36
2.3. Використання технологій в розроблюваній автоматизованій системі .....	40
2.4. Структурний розрахунок приладу.....	43
2.5. Висновки до розділу .....	47
3. Опис роботи програми.....	49
3.1. Використані бібліотеки .....	49
3.2. Код програми.....	50
3.3. Структурний аналіз програмного забезпечення .....	68
3.4. Використання програми .....	71
3.5. Висновки до розділу .....	73
4. РОЗРОБКА СТАРТАП-ПРОЄКТУ «АВТОМАТИЗОВАНИЙ ПОРТАТИВНИЙ ПРИЛАД ДЛЯ ВИМІРЮВАННЯ НАПРУЖЕНОСТІ ЕЛЕКТРОСТАТИЧНОГО ПОЛЯ НА МІСЦЕВОСТІ» .....	74
4.1. Опис технологічний аудит ідеї проєкту.....	74
4.2. Аналіз ринкових можливостей запуску стартап-проєкту .....	77



4.3. Розроблення ринкової стратегії продукту .....	84
4.4. . Розроблення маркетингової програми стартап-проєкту .....	86
4.5. Організація реалізації стартап-проєкту .....	89
4.6. Висновки до розділу .....	94
<b>Висновки .....</b>	<b>95</b>
Список використаних джерел .....	96

## **Перелік основних скорочень**

НК – неруйнівний контроль;

АЦП – аналого-цифровий перетворювач;

ЦАП – цифро-аналоговий перетворювач;

ПК – персональний комп'ютер;

ПЗ – програмне забезпечення.

## ВСТУП

Задача вимірювання напружених електричних полів впливає на різні промисловості, пов'язані з виробництвом або з широким використанням синтетичних матеріалів, де може бути використана статична електричність і що може привести до серйозних наслідків.

Також з давніх часів вчені цікавились вимірюванням електростатичних полів атмосфери.

Розширення географії та збільшення інтенсивного вимірювання атмосферного електричного середовища привели до того, що було виявлено зв'язок з геомагнітними аномаліями земної кори. Це відкриття цікавить геофізиків, використовуючи його для вирішення завдань пошуку та розвідки корисних копалин, а також вирішення задач інженерної геології.

Складність вимірювань електричних полів атмосфери полягає в дуже маленьких значень даних електричних полів. Саме тому потрібно використовувати спеціальне обладнання, і спеціальні методи вимірювань.

Більшість існуючих методів, не досконалі, та мають свою проблематику. Саме тому задачею даної магістерської роботи, є вдосконалення одного з найкращих методів вимірювання напруженості електричного поля атмосфери, а саме динамічного методу.

Також електричне поле атмосфери, є недостатньо вивченим в інших сферах. Наприклад в сфері впливу різних значень електричного поля атмосфери на людину.

Автоматизованих систем для виміру електричного поля атмосфери на даний момент зовсім не існує, а це могло б сильно спростити, процес геодезичних розслідувань, а також процес дослідження впливу значень електричного поля атмосфери в різних сферах.

Саме тому головною задачею даної роботи є створення автоматизованої системи вимірювання напруженості електричного поля атмосфери на місцевості,

а також програмного забезпечення, яке після обробки результатів вимірювання подавало б їх в зручному вигляді.

## 1. Аналітичний огляд

### 1.1. Електричне поле в атмосфері

Провівши через точки з рівним значенням потенціалу неперервні поверхні, отримаємо рівневі, або ізо-потенціальні поверхні. Поверхні в силу відомих нам властивостей потенціалу не мають змоги ні перетинатися один з одним, ані торкатися.

Якщо взяти в будь якій точці рівномірної поверхні похідну від потенціалу відносно нормалі до неї  $n$ , то отримаємо для точки повну напруженість поля:

$$E = -\frac{dV}{dn} = -gradV \quad (1.1)$$

Тому звідси випливає, що в місцях, де рівень поверхні наближається один до одного, напруженість поля буде по абсолютній величині більше, і навпаки. Рівневі поверхні майже слідує за рельєфом земної поверхні, наближуючись над усіма височинами і розходячись над поглибленням. Зі зміною висоти їх хід потрохи вирівнюється.

Безпосередньо недалеко від земної поверхні, що є провідником, для напруженості поля будемо мати співвідношення:

$$E_0 = -\frac{dV}{dh} = \frac{\sigma}{\epsilon_0 \epsilon} \quad (1.2)$$

де  $a$  – густина поверхневого заряду в даній точці.

За наявності в атмосфері об'ємних зарядів у кожній точці атмосфери повинно бути виконано рівняння Пуассона:

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = -\frac{1}{\epsilon_0 \epsilon} \rho \quad (1.3)$$

З формули (1.1) можемо побачити, що напруженість поля влюбій точці за числом дорівнює зміні потенціалу, який розрахований на одиницю відстані за напрямку нормалі до рівномірної поверхні, яка проведена в сторону спадаючого потенціалу. А на практиці напруженість поля і за числом рівний їй, але

обернений за знаком, градієнт потенціалу буде вимірюватись в вольтах на метр (В / м).

Звичайно стосовно до атмосфери зазначені співвідношення використовуються лише з якимось наближенням. Дійсно, хоч атмосфера і є середовищем неоднорідним в електричному відношенні, але в більшості випадків неоднорідність атмосфери в горизонтальному напрямку сильно менше, ніж у вертикальному. Це надає можливість при першому наближенні знехтувати врахуванням горизонтальних неоднорідностей і ми можемо розглядати поле що знаходиться над великою плоскою рівниною якби однорідне. Для того щоб описати поля в атмосфері потрібно спростити рівняння (1.1) і (1.3), замінивши нормаль і координату  $z$  через висоту  $h$  і поклавши  $\frac{d^2 V}{dx^2} = \frac{d^2 V}{dy^2} = 0$ . Тоді:

$$\frac{d^2 V}{dh^2} = -\frac{dE}{dh} = -\frac{1}{\epsilon_0 \epsilon} \rho \quad (1.4)$$

Спостереження, які були проведені в земній поверхні, дають в середньому градієнта потенціалу значення близько 130 В / м. Застосувавши рівняння (1.2), маємо змогу обчислити середнє значення поверхневої густини об'ємного заряду Землі  $\sigma$ , яка дорівнює приблизно  $-1,15 * 10^9$  Кл/м<sup>2</sup>. Тому як поверхня Землі дорівнює  $4\pi R^2$ , де  $R$  - радіус Земної кулі, то заряд Землі буде дорівнювати  $Q = 4\pi R^2 \sigma = -5,7 * 10^5$  Кл.

Напруженість електричного поля атмосфери має змогу бути розрахованою за вимірним значенням вертикального струму провідності атмосфери і відповідним значенням провідності  $\lambda$ :

$$E = \frac{i}{\lambda} \quad (1.5)$$

Струм провідності атмосфери як правило не змінюється з висотою. Отже, маємо що напруженість електричного поля напряму залежить, по-перше, від самої провідності атмосфери, а по-друге, від самих об'ємних зарядів в атмосфері, які локалізовані в досліджуваній області. Саме тому зміни напруженості поля

напряму відповідають зміні провідності, але на це ще накладається вплив не однорідність об'ємних зарядів атмосфери.

На рисунку представлена схема, яка демонструє формування електричного поля в атмосфері Землі і електричних струмів в біосфері також показано, які космічні і земні чинники можуть значно впливати на ці питання.

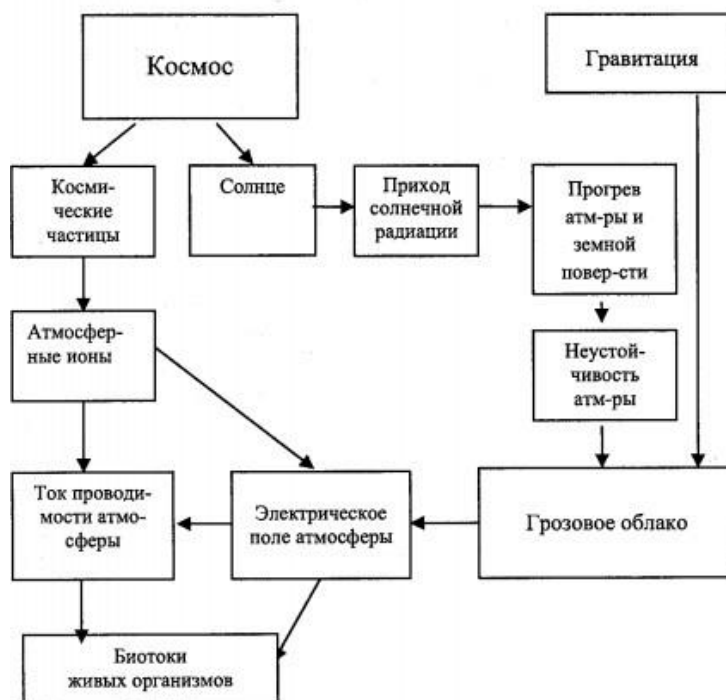


Рис. 1.1 - Схема формування електричного поля атмосфери Землі і електричних струмів в біосфері.

## 1.2. Динамічний метод вимірювання напруженості електричного поля в атмосфері

Існує два основних методи вимірювання напруженості електричного поля в атмосфері:

- Статичний
- Динамічний

Статичний метод потребує великого проміжку часу для виконання заміру

на відміну від динамічного способу, який є більш новим, та швидшим. Тому в даній роботі буде розглянуто датчик який працює на основі динамічного метода. Розглянемо принцип виміру даним методом.

Динамічний метод вимірювання напруженості електричного поля заснований на застосуванні електростатичних генераторів. Ці останні є окремим випадком більш загального вигляду генераторів, так званих параметричних генераторів, загальна теорія яких була розроблена Л. І. Мандельштамом і М. Д. Папалексі.

Дія електростатичного генератора базується на тому, що при внесенні в змінне електричне поле провідного тіла в останньому виникає рух індукованих зарядів. Величина струму, створюваного зарядами що переміщуються, пропорційна, що викликає ці зміни. Ця обставина використовується для створення приладів, що вимірюють напруженість поля.

Існує багато конструкцій електростатичних генераторів, які застосовуються для цілей вимірювання поля.

Будь-яка конструкція генератора повинна перетворювати вимірюване постійне електричне поле в змінне, що впливає на вимірювальний електрод. У всіх цих конструкціях існує елемент, за рахунок періодичного переміщення якого змінюється величина електричного поля на електроді, що сприймає поле.

Роботу електростатичного генератора зручно описати, користуючись схемою, представленої на Рис. 1.2 . Ця схема дозволяє зручно проводити вимірювання електричного поля  $i$ , разом з тим, є досить типовою,

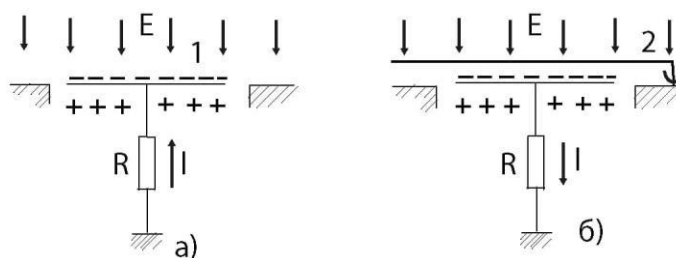


Рис. 1.2 - Динамічний метод вимірювання напруженості поля.



щоб з опису її дії можна було перейти до розрахунку будь-якого іншого варіанту електростатичного генератора.

Плоска металева пластина 1 (яку аналогічно до попереднього назвемо вимірювальною пластиною) площею  $S_0$  з'єднана з землею через опір  $R$ , вимірювальна пластина може бути експонована в поле  $E$ , нормальному до пластини (Рис. 1.2 , а), або ж бути закритою від нього з допомогою обертового заземленого екрану 2 (Рис. 1.2 , б). При цьому по опорі буде текти струм.

Якщо обертання екрану здійснювати безперервно, то по опорі  $R$  буде весь час текти змінний струм  $i$ .

Позначимо через  $S$  частину площі вимірювальної пластини, відкритої в певний момент часу  $t$  дії поля, і через  $C$ -ємність вимірювальної пластини по відношенню до землі (ця величина буде також, очевидно, змінюватися в часі). Тоді струм, що протікає по опорі  $R$  і створить падіння напруги  $R_i = R \frac{dQ}{dt}$ , де  $Q$  – заряд пластини. З другого боку, струм по опорі  $R$  створюється за рахунок різниці потенціалів між вимірювальною пластиною і землею.

Потенціал вимірювальної пластини в будь-якої момент часу  $t$  складається з двох частин: з потенціалу який набуває металева пластина, внесена в електричне поле (цей потенціал пов'язаний з напруженістю поля  $E$  співвідношенням  $V_1 = \frac{ES}{4\pi C}$ ), і з потенціалу  $V_2$  який створює заряд  $Q$ , що з'явився за рахунок струму, який протікає по опорі  $R$  на пластину. Цей потенціал дорівнює  $V_2 = -\frac{Q}{C}$  Очевидно, що потенціал, створюваний зарядом  $Q$ , по знаку протилежний потенціалу, створюваному полем.

Ми можемо, відповідно, написати  $V_1 - V_2 = iR$ , або:

$$\frac{ES}{4\pi C} - \frac{Q}{C} = R \frac{dQ}{dt} \quad (1.6)$$

Поле також мінливе в часі:  $E = E(t)$ , можемо переписати рівняння (1.6):

$$\frac{dQ}{dt} + \frac{Q}{RC} - \frac{ES}{4\pi RC} = 0 \quad (1.7)$$

Вирішення цього рівняння дає:

$$Q = e^{-\int \frac{d}{RC}} \left( \int \frac{ES}{4\pi RC} e^{\int \frac{dt}{RC}} dt + B \right) \quad (1.8)$$

Значення струму і можна обчислити з виразу:

$$i = \frac{dQ}{dt} \quad (1.9)$$

Значення довільної сталої  $B$  при встановленому режимі визначається з умови ненакопичуваного заряду, тобто з умови, що зміна  $Q$  за весь цикл має дорівнювати нулю.

Рівняння (1.6), що враховує як зміну відкритої частини площі вимірювальної пластини  $S$ , так і визначення напруженості поля  $E$  і ємності  $C$ , є рівнянням, придатним для будь-якого електростатичного генератора, оскільки на зміні цих трьох елементів (або деяких з них) ґрунтується робота всіх відомих електростатичних генераторів. Рішення (1.8) в силу цього дає можливість розрахунку будь-якого електростатичного генератора, якщо задані його параметри.

Рівняння (1.8) демонструє також можливість застосування електростатичного генератора для вимірювання напруженості постійних електричних полів. Якщо  $E = \text{const}$ , то між  $Q$  і  $E$  існує лінійна залежність, та ж залежність буде, очевидно, існувати для зв'язку  $i$  з  $E$ . Ця залежність буде залишатися до тих пір, поки  $E = \text{const}$ , або поки за час одного циклу роботи напруженість вимірюваного поля  $E$  встигне помітно змінитися.

Рішення (1.8) вказує також на можливість вимірювання полів, мінливих під час одного циклу, але в цьому випадку залежність  $i$  від  $E$  не має вже лінійного характеру.

Рішення (1.8) внаслідок його громіздкою форми важко повністю проаналізувати в загальному вигляді. Аналіз роботи електростатичного генератора зручніше провести на певній схемі, оскільки рівняння які виходять в цьому випадку дозволяють детально розглянути процеси при роботі.

Повернемося до розглянутої схеми Рис. 1.2 .

Якщо відкрита частина пластини  $S$  лінійно змінюється в часі (наприклад вимірювальна пластина і екран мають форму півкіл або рівнокутних секторів і екран рівномірно обертається-випадок, зустрічається найчастіше), то  $C$  також буде лінійно змінюватися в часі. Зміни  $S$  і  $C$  в часі представлені на наступному Рис. 1.3 (криві а й б відповідно).

Щоб вирішити задачу про зв'язок і с  $E$  і з параметрами генератора, рішення рівняння (1.6) будемо вести окремо для проміжку часу, коли вимірювальна пластина відкривається, і для проміжку часу, коли вона закривається.

1) Вимірювальна пластина відкривається.

В цьому випадку можна написати:

$$S = S_0 \frac{\omega}{\pi} t \quad (1.10)$$

Де  $\omega$  – кутова швидкість обертання екрана, або  $S = pt$ , де  $p = \frac{S_0}{\pi} \omega$

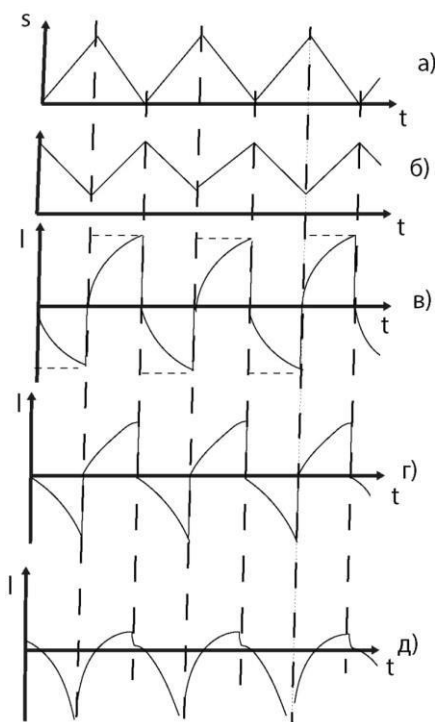


Рис. 1.3 - Струм і напруга на виході електростатичного генератора в залежності від періоду обертання екрануючої пластини (струмовий режим).

Ємність вимірювальної пластини змінюється при обертанні екрана від максимального значення  $C_{\text{макс}}$  до мінімального  $C_0$  і

$$C = C_{\text{макс}} - \frac{C_{\text{макс}} - C_0}{\pi} \omega t = C_{\text{макс}} - k \quad (1.11)$$

Де:

$$k = \frac{C_{\text{макс}} - C_0}{\pi} \omega \quad (1.12)$$

Повертаючись до рішення рівняння(1.8) і вважаючи  $E = \text{const}$ , ми отримаємо:

$$Q_0 = -\frac{Ep}{4\pi} \left[ \frac{t - RC_{\text{макс}}}{kR - 1} + B \left( 1 - \frac{k}{C_{\text{макс}}} t \right)^{\frac{1}{kR}} \right] \quad (1.13)$$

де  $B$  – постійна інтегрування.

Індекс при  $Q_0$  означає, що пластинка відкривається.

Значення струму і можна знайти із рівності:

$$i_0 = \frac{ES_0\omega}{4\pi^2} \left[ \frac{1}{kR - 1} + \frac{B}{RC_{\text{макс}}} \left( 1 - \frac{k}{C_{\text{макс}}} t \right)^{\frac{1}{kR} - 1} \right] \quad (1.14)$$

Розглянемо другу частину циклу.

2) Вимірювальна пластинка закривається.

В цьому випадку  $S = S_0 - pt$  і:

$$C = C_0 + \frac{C_{\text{макс}} - C_0}{\pi} \omega t = C_0 + k \quad (1.15)$$

Повертаючись знову до вирішення рівняння (1.8), отримаємо, вважаючи  $E = \text{const}$ :

$$Q_a = -\frac{Ep}{4\pi} \left[ -\frac{\pi}{\omega} + \frac{t - RC_0}{kR + 1} + D \left( 1 + \frac{k}{C_0} t \right)^{-\frac{1}{kR} - 1} \right] \quad (1.16)$$

при чому індекс  $Q_a$  означає, що пластина закривається. Для струму  $i_a$  знайдемо:

$$i_a = \frac{ES\omega}{4\pi^2} \left[ \frac{1}{kR + 1} - \frac{D}{RC_0} \left( 1 + \frac{k}{C_0} t \right)^{-\frac{1}{kR} - 1} \right] \quad (1.17)$$

де  $D$  - постійна інтегрування.

Значення постійних  $B$  і  $D$  можна обчислити з наступних умов: 1) заряд пластини є неперервною функцією часу; 2) заряди на початку однієї половини циклу і в кінці іншої рівні між собою; 3) не відбувається накопичення заряду за весь цикл.

Для спрощення отриманих виразів введемо деякі додаткові позначення: позначимо період процесу через  $r = \frac{2\pi}{\omega}$ ; тимчасову постійну схеми в момент повного екранування пластини через  $r_{\text{макс}} = RC_{\text{макс}}$  і тимчасову постійну схеми в момент повного експонування пластини через  $r_0 = RC_0$ . Тоді вираз для постійних набуває наступного вигляду:

$$B = \frac{2r_{\text{макс}} - 2r_0 \left( \frac{C_0}{C_{\text{макс}}} \right)^{\frac{1}{kR}}}{\left[ 4 \left( r_{\text{макс}} - r_0 \right)^2 - r_0^2 \right] \left[ 1 - \left( \frac{C_0}{C_{\text{макс}}} \right)^{\frac{2}{kR}} \right]} r^2, \quad (1.18)$$

$$D = \frac{2r_{\text{макс}} \left( \frac{C_0}{C_{\text{макс}}} \right)^{\frac{1}{kR}} - 2r_0}{\left[ 4 \left( r_{\text{макс}} - r_0 \right)^2 - r_0^2 \right] \left[ 1 - \left( \frac{C_0}{C_{\text{макс}}} \right)^{\frac{2}{kR}} \right]} r^2, \quad (1.19)$$

Незважаючи на деяку громіздкість коефіцієнтів, можна провести повний розрахунок залежності і від  $E$ , знаючи параметри схеми, або ж можна визначити

необхідні параметри схеми по встановленим параметрам і при заданому Е. Підставляючи значення В і D з рівнянь (1.18) і (1.19) в рівняння (1.14) і (1.15) відповідно, отримуємо рівняння:

$$i_0 = \frac{ES\omega}{4\pi^2(kR-1)} * \left( 1 - 2 \frac{1 - \left(\frac{C_0}{C_{\max}}\right)^{\frac{1}{kR+1}}}{(kR+1) \left[1 - \left(\frac{C_0}{C_{\max}}\right)^{\frac{2}{kR}}\right]} \left(1 - \frac{k}{C_{\max}} t\right)^{\frac{1}{kR}-1} \right) \quad (1.20)$$

$$i_0 = \frac{ES\omega}{4\pi^2(kR+1)} * \left( 1 - 2 \frac{\left(\frac{C_0}{C_{\max}}\right)^{\frac{1}{kR-1}} - 1}{(kR-1) \left[1 - \left(\frac{C_0}{C_{\max}}\right)^{\frac{2}{kR}}\right]} \left(1 + \frac{k}{C_0} t\right)^{-\frac{1}{kR}-1} \right) \quad (1.21)$$

Слід звернути особливу увагу на параметр:

$$kR = \frac{C_{\max} - C_0}{\pi} \omega R, \quad (1.22)$$

або в введених позначеннях:

$$kR = 2 \frac{r_{\max} - r_0}{r} \quad (1.23)$$

Величина  $kR$  показує відношення зміни часової постійної схеми до часу одної половини циклу. Залежно від величини цього параметра змінюється характер залежності  $i = f(i)$ . На Рис. 1.3, в і г залежність  $i = f(t)$  представлена графічно для  $kR < 1$ .

У цьому випадку струм під час відкривання пластини може бути виражений рівнянням виду:

$$i_0 = -|\alpha| + |\delta| \left(1 + \frac{k}{C_{\max}} t\right)^{\frac{1}{kR}-1}, \quad (1.24)$$

а під час закривання:

$$i_a = |\beta| - \frac{|\chi|}{\left(1 + \frac{k}{C_0} t\right)^{\frac{1}{kR}+1}} \quad (1.25)$$

Поки  $kR < \frac{1}{2}$ , величини  $i_0$  і  $i_a$  мають вигляд, подібний представленому на,

Рис. 1.3 в, причому з зменшенням  $kR$ , коли період обертання стає набагато більше зміни тимчасової постійної схеми, форма  $i_0$  і  $i_a$  наближається до прямокутної (на рисунку показано пунктирною лінією); при цьому амплітудне значення напруги на виході генератора падає.

Якщо  $\frac{1}{2} < kR < 1$ , то форма  $i_0$  змінюється, як показано на Рис. 1.3, г, переходячи через лінійну залежність при  $kR = \frac{1}{2}$  (на рисунку показано пунктирною лінією). При  $kR = \frac{1}{2}$  спадання потенціалу внаслідок стікання заряду і зростання його внаслідок зменшення ємності системи призводять до лінійної залежності струму від часу. При  $kR = \frac{1}{2}$  рівняння кривої струму при процесі відкриття пластини набуває вигляду:

$$i_0 = |\alpha| - \frac{|x|}{\left(1 + \frac{k}{C_{\max}} t\right)^{1 - \frac{1}{kR}}} \quad (1.26)$$

При закриванні пластини, рівняння кривої струму зберігає вид, що дається рівнянням (1.25). Форма кривої струму в цьому випадку представлена на Рис. 1.3, д. Видно зрушення струму по фазі: крім того, внаслідок великої тимчасової постійної схеми заряд не встигає повністю стікати, що призводить до зменшення струму, що тече по опорі.

Як видно з наведених рівнянн і малюнків, збільшення числа оборотів призводить до зростання ефективного значення струму тільки поки  $kR < \frac{1}{2}$ . Це відношення обмежує можливість підвищення чутливості схеми за рахунок збільшення числа оборотів.

Зміна знаку вимірюваного поля змінює в усіх випадках напрямок протікання струмів на протилежне.

Розглянемо ще один варіант використання електростатичного генератора, в якому величина поля визначається за допомогою вимірювання потенціалу вимірювальної пластини, причому величина  $kR$  дуже велика ( $R$  дуже велике) і  $R \gg \frac{1}{\omega C_0}$ . У цьому випадку на вимірювальній пластині можуть перебувати тільки індуковані заряди.

Потенціал пластини  $V$  тоді можна знайти з рівняння:

$$V = \frac{ES}{4\pi C} \quad (1.27)$$

Зміні  $S$  і  $C$  (Рис. 1.4 , а і б) відповідає залежність потенціалу пластини від часу представлена на Рис. 1.4 , в і г. На Рис. 1.4 , в, показано зміна потенціалу при  $C \rightarrow 0$ , а на Рис. 1.4 , в і г, зображений випадковий, коли  $\frac{C_{\max} - C_0}{C_{\max}} \rightarrow 0$ . Як видно з рисунків, для збільшення чутливості схеми вигідно брати  $C_0$  можливо меншим, але це призводить до вимоги надзвичайно малої інерційності вимірювального приладу, оскільки основне зростання потенціалу припадає на малу частину часу циклу. Збільшення  $C_0$  робить наростання потенціалу більш плавним, але зате знижує амплітудне значення потенціалу.



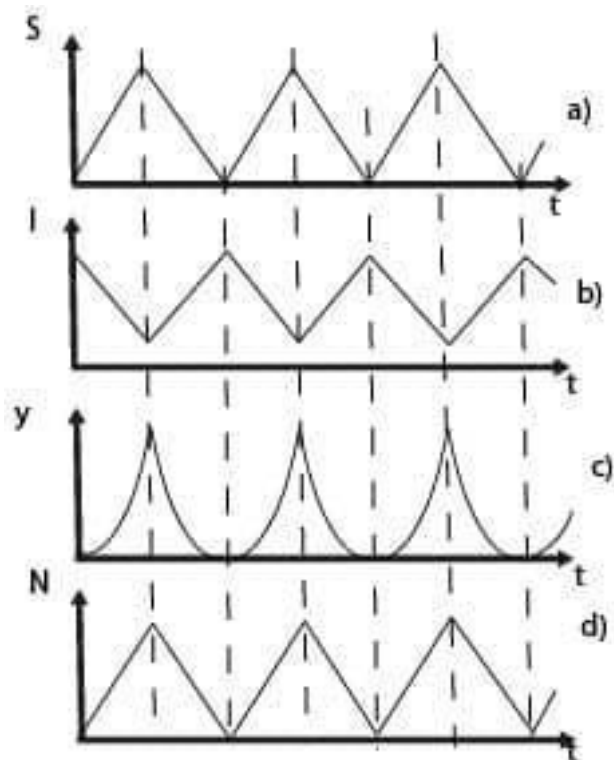


Рис. 1.4 - Напряга на виході електростатичного генератора в залежності від періоду обертання екрануючої пластини (зарядний режим).

Хороші результати дає вимір напруги на пластині за допомогою електрометричної лампи, причому вимірюють напругу на анодному навантаженні цієї лампи за допомогою катодного осцилографа. Таким чином, вдається поєднати малу початкову ємність і без інерційний вимір амплітуди напруги. Ця схема дозволяє значно знизити межу вимірюваних напруженостей поля.

Прилад для вимірювання заряду окремих частинок, заснований на цьому методі, в зв'язку з вимірюванням величини зарядів частинок в атмосфері.

Порівнюючи чисто струмовий режим, коли заряд вимірювальної пластини за час циклу встигає повністю стікати ( $R \ll \frac{1}{\omega C_0}$ ), і режим напруги, коли на вимірювальній пластині є тільки індукований заряд ( $R \gg \frac{1}{\omega C_0}$ ), можна встановити наступне (Рис. 1.3 і Рис. 1.4).

Перехід від токового режиму до режиму напруги супроводжується зрушенням фази напруги на виході генератора на  $90^\circ$ .

Далі, при роботі в струмовому режимі на виході генератора виходить змінний струм; зміна знаку вимірюваного поля зрушує фазу цього струму на  $180^\circ$ . При роботі ж в режимі напруги на виході генератора виходить постійна пульсуюча напруга; вимірювання знаку вимірюваного поля змінює знак цієї напруги.

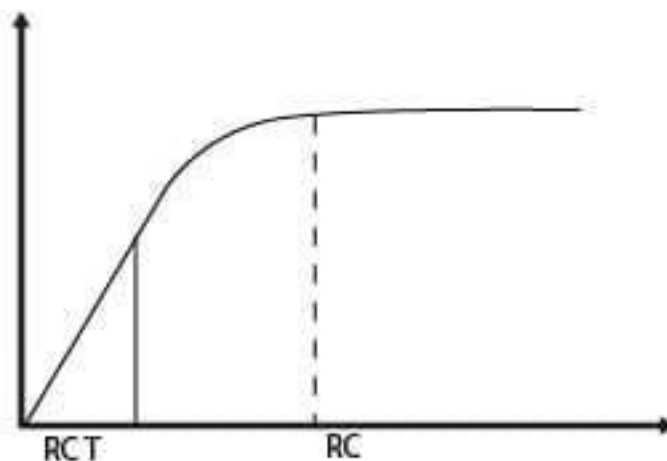


Рис. 1.5. - Залежність середньої напруги  $V$  на виході електростатичного генератора від відношення часової постійної  $RC$  схеми до періоду обертання екрану  $t$ .

Повертаючись до вимірювань напруженості поля за величиною струму, поточного з пластини, або по її потенціалу, слід зазначити, що майже всі прилади, побудовані за цим принципом, вимірюючи не миттєве значення струмів або потенціалів, а величину середнього струму, поточного по опорі  $R$ , або середнє значення потенціалу пластини. Цей середній струм (або потенціал) або вимірюється безпосередньо приладом після попереднього випрямлення (оскільки безпосереднє вимірювання малих змінних струмів або напруг ускладнене), або цей струм (або потенціал) посилюється і випрямляється ламповою схемою, а потім вимірюється.

Починаючи від тих значень  $RC_0$  коли за час  $t$  на пластину не встигають натекти заряди, амплітуда і середнє значення потенціалу на пластинах вже практично не залежать від величини  $r$ . На рисунку 1.8 дана залежність середньої напруги  $V$  на виході генератора від швидкості обертання екрану  $\omega$ . В тих випадках коли мотор, що обертає екран, не може підтримувати постійне число

обертів, для підвищення точності вимірювання генератор змушують працювати в області  $RC_0 \gg r$ .

Розглядаючи рівняння (1.20) (1.21) (1.26), приходимо до наступних висновків про використання електростатичного генератора в якості вимірювального приладу.

1) При вимірах напруженості поля по амплітудним значенням напруги треба прагнути до значень  $kR \gg 1$  за рахунок збільшення  $R$  при мінімальному  $C_0$ . В цьому випадку необхідно ставити на виході генератора без інерційний вимірювальний прилад.

2) Для вимірювань поля за середнім струмом, що протікає з вимірювальної пластини, щоб апарат знаходився в області значень  $kR < \frac{1}{2}$  при  $RC_0 \ll r$ .

3) Застосування електростатичного генератора в поєднанні з ламповим підсилювачем призводить до потреби в максимальній потужності на виході генератора. У цьому випадку апарат знаходився в області  $1 > kR > \frac{1}{2}$ . Вибираючи в якості першої лампи підсилювача лампу з малими сітковими струмами, можна, щоб виключити вплив частоти обертання екрану, працювати в області:

$$RC_0 = 5 \div 10r \quad (1.28)$$

Для  $kR \ll 1$  і  $RC_0 \ll r$  в рівняннях (1.14) і (1.17) можна знехтувати другим членом в дужках, де враховується вплив мінливої ємності. Рівняння (1.14) і (1.17) можна в цьому випадку представити одним рівнянням:

$$i = \pm \frac{ES_0\omega}{4\pi^2} \quad (1.29)$$

причому першу половину циклу струм тече в одному напрямлені, а другу половину - в іншому (Рис. 1.2 , в).

Слід зазначити, що у випадках, коли бажано виміряти не тільки величину, а й знак напруженості, в прилад часто вводять спеціальний пристрій,

синхронізуючий з екраном, що дозволяє визначити фазу струму або напруги і тим самим і напрямок поля.

В електростатичному генераторі енергія, що виділяється на опорі  $R$ , поставляється роботою механічних сил, витрачених на переміщення екрану, без витрати енергії електричного поля. Остання обставина є головною перевагою методу електростатичних генераторів, дозволяючи оперувати при вимірюванні струмами і потужностей, значно вищими, ніж при інших методах вимірювання.

Однак струми, що виходять на виході електростатичного генератора, виявляються достатніми для безпосереднього вимірювання чуттєвим гальванометром тільки в порівняно великих полях, причому в схему необхідно включити випрямний пристрій. Так як всі лампові і напівпровідникові випрямлячі погано працюють при малих напругах, часто вдаються до механічних випрямлячів; проте це дуже ускладнює схему, а надійність роботи механічних випрямлячів зазвичай недостатня.

Якщо потрібно вимірювати малі напруженості електричного поля і бажано обійтися без надмірно чуттєвих гальванометрів на виході, в прилад доводиться вводити лампову підсилювальну схему. Наявність змінної напруги на виході генератора надто полегшує посилення і дозволяє отримати досить високе відношення сигналу до шуму.

Другим фактором, що збільшує відношення сигналу до шуму, є можливість працювати з вхідними опором порядку  $10^3 - 10^4$  ом (замість опору, що перевищують  $10^{13} - 10^{14}$  ом, звичайних в електрометричних схемах для вимірювання напруженості поля); це дозволяє значно знизити вплив наведень. Низькі вхідні опори також надзвичайно зменшують вплив дефектів ізоляції на роботу приладу.

Потужність, що дається електростатичним генератором, виявляється достатньою для того, щоб вимірювати поля напруженістю до  $0,001$  е/см за допомогою підсилювачів на звичайних серійних радіолампах, що працюють в звичайних режимах.

Таким чином, використовуючи схему електростатичного генератора з підсилювачем, можна вимірювати дуже малі напруженості електричного поля. При правильному виборі схеми вдається довести інерцію цих приладів до тисячних часток секунди.

### 1.3. Схема та принцип роботи датчику

А тепер розглянемо принцип роботи самого датчика та його структурну схему датчика, який працює на базі динамічного методу

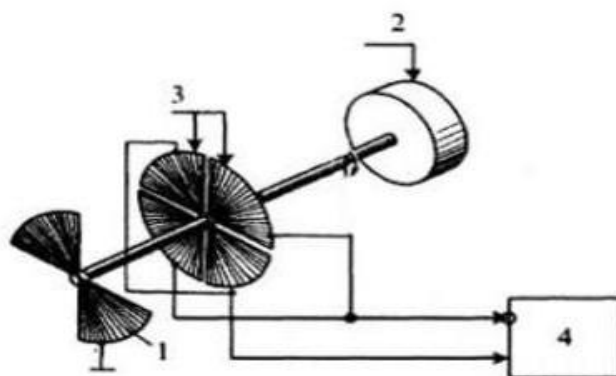


Рис. 1.4- Блок схема вимірювального перетворювача динамічного(ротаційного) типу

Пластина під номером 1, що заземлена на корпус, поперемінно починає екранувати і експонувати дві пари вимірювальних пластин під номером 3, що підключені до входу нашого підсилювача під номером 4 і відповідно до цього в самих пластинах змінюється індукований заряд, що викликає змінний струм в резисторах під маркуванням R, які умовно визначені двома диференціальними входами на підсилювачі під номером 4. Не є важким показати, що за період обертання  $T$  пластини, яка заземлена під номером 1, яка демонструється на рис. 1.5 струм буде змінювати свій напрямок 4 рази. З цього можна зробити висновок, що розглянутий нами на рисунку 1.5 динамічний метод вимірювання може дозволити виконувати одне повноцінне вимірювання напруженості електричного поля за відрізок часу, що не буде більшим ніж півперіод обертання

двигуна, що є в декілька порядків швидше ніж час вимірювань статичним методом.

Тому підсумувавши все що було сказано вище, а саме про високу чутливість та маленький час виміру, для реалізації поставлених нами вимог до вимірювача напруженості електростатичних полів доцільніше буде використовувати динамічний метод вимірювання.

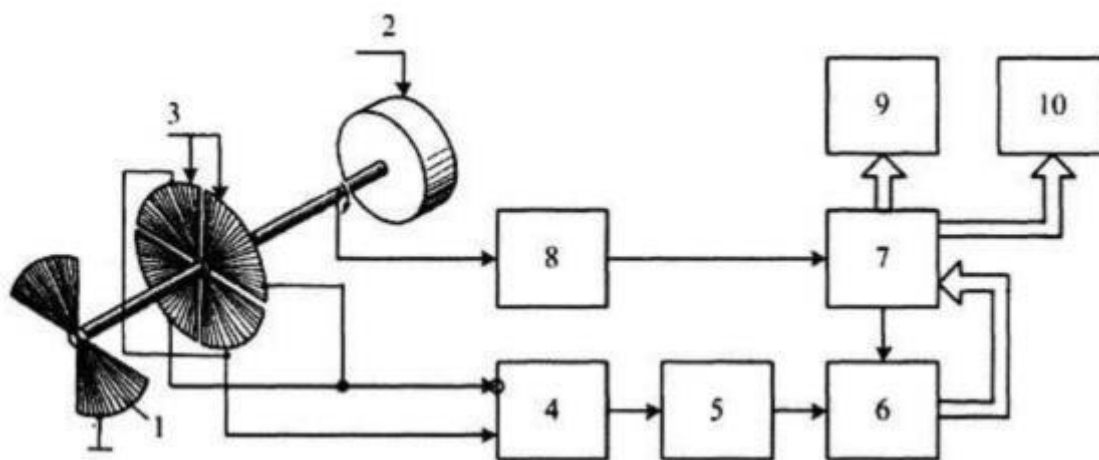


Рис. 1.5 – Функціональна схема вимірювача напруженості електростатичного поля:

1 – модулятор; 2 – електродвигун; 3 – вимірювальні пластини; 4 – підсилювач; 5 – фільтр; 6 – АЦП; 7 – мікроконтролер; 8 – датчик синхронізації положення пластин; 9 – дисплей; 10 – модуль безпроводної передачі сигналу.

#### 1.4. Висновки до розділу

В даному розділі був проведений огляд об'єкта вимірювання, а саме електричного поля атмосфери. Також розглянутий динамічний метод вимірювання напруженості електричного поля атмосфери, що базується на електростатичних генераторах. Даний метод є поки що одним з найкращих, так як відрізняється від статичного швидким часом виміру, декілька десятих секунди при використанні динамічного методу, і як мінімум декілька годин при

використані статичного методу вимірювання. Також динамічний метод є більш точним. Також в даному розділі був проведений огляд самого вимірювача, і принципу його роботи, що заснований на динамічному методі.

## 2. Розробка автоматизованої системи

Для розробки автоматизованої системи виміру напруженості електричного поля на місцевості, до нашого датчика потрібно підключити GPS-модуль, та систему пам'яті, в даному випадку була обрана SD-Card

### 2.1. Технологія GPS

GPS (глобальна система позиціонування) — це супутникова навігаційна система, що складається щонайменше з системи супутників. Ці супутники обертаються навколо Землі на висоті близько 20 000 км. І здійснюють два повних оберти навколо землі за день. Він працює 24//7 по всьому світу незалежно від погодних умов. За його використання також не стягується плата за налаштування чи передплата.

GPS - це система з трьох частин, яка включає:

- a) Супутники
- b) Наземні станції – вони контролюють і керують супутниками. Наземні станції також визначають своє місце розташування.
- c) Приймачі – приймачі постійно прослуховують сигнали від супутників.

Система була розроблена військовими США. Вони впровадили цю супутникову мережу як військову навігаційну систему, але незабаром відкрили її для всіх інших.

Кожен з супутників працює на сонячних батареях. Орбіти влаштовані так, що в будь-який момент і в будь-якій точці Землі на небі «видимі» щонайменше чотири супутники.

Завдання приймача GPS полягає в тому, щоб знайти чотири або більше цих супутників, визначити відстань до кожного з них і використати цю інформацію для визначення власного місцезнаходження. Ця операція заснована на простому математичному принципі, який називається тріангуляція. тріангуляція в тривимірному просторі може бути трохи складною, тому ми почнемо розгляд з простої двовимірної тріангуляції.



2D триангуляція(або метод трьох базових станцій) – це метод, що дозволяє знайти, точку на карті на якій знаходиться користувач, за допомогою трьох або більше базових станцій.

Цей метод полягає в тому, що ми маємо три, або більше базові станції, і знаємо точне їх місце знаходження. Знаючи відстань від однієї з таких базових станцій ми можемо намалювати радіус з даним радіусом, що означає, що користувач може знаходитись будь де на цьому радіусу. Знаючи відстань від другої базової станції, ми можемо побудувати, ще одне коло, і отримуємо дві точки перетину, цих двох кіл, користувач може знаходитись будь де на цьому колі. І коли буде відомо відстань від третьої базової станції, ми матимемо змогу побудувати 3 радіус, і в результаті ми отримаємо одну точку, де будуть перетинатись всі три кола, ця точка, і буде точкою знаходження нашого користувача.



Рис. 2.1 – 2D триангуляція

### 3D триангуляція

По суті, тривимірна трилатерація мало чим відрізняється від двовимірної трилатерації, але її трохи складніше уявити. Уявіть, що радіуси з попередніх прикладів розходяться в усіх напрямках. Отже, замість серії кіл ви отримаєте серію сфер.

Якщо ви знаєте, що перебуваєте в 10 кілометрах від супутника А в небі, ви можете опинитися в будь-якому місці на поверхні величезної уявної кулі з радіусом 10 кілометрів. Якщо ви також знаєте, що знаходитесь на відстані 15 кілометрів від супутника В, ви можете перекрити першу сферу іншою, більшою. Кулі перетинаються в ідеальному колі. Якщо ви знаєте відстань до третього супутника, ви отримаєте третю кулю, яка перетинається з цим колом у двох точках.

Сама Земля може діяти як четверта куля - лише одна з двох можливих точок буде фактично на поверхні планети, тому ви можете усунути ту в космосі. Однак приймачі зазвичай звертаються до чотирьох або більше супутників, щоб підвищити точність і надати точну інформацію про висоту.

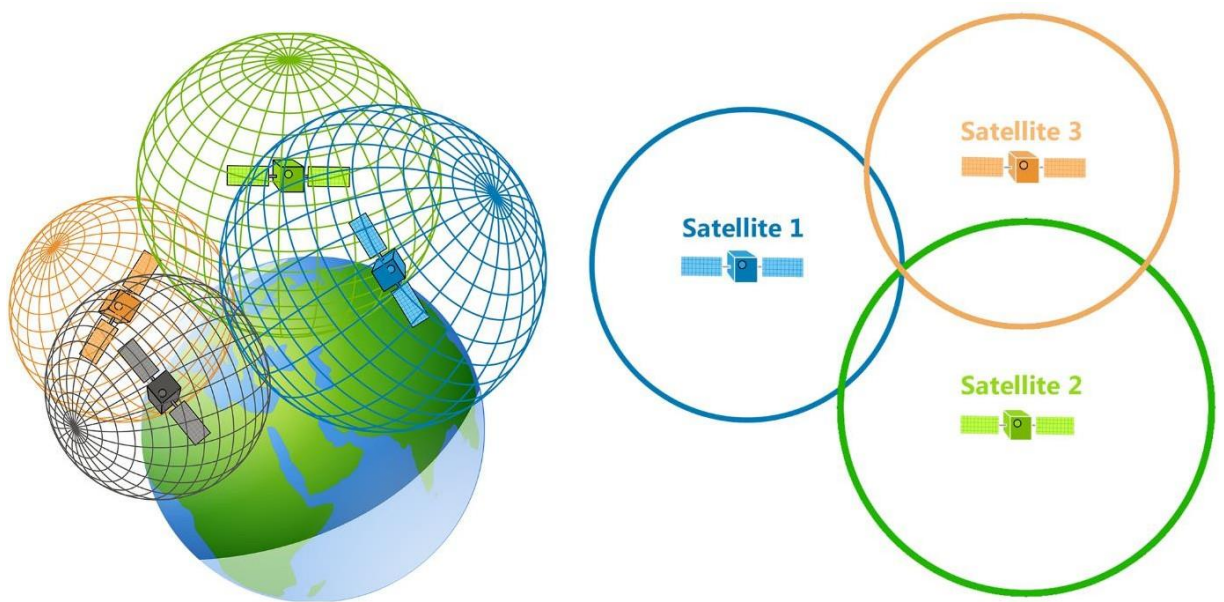


Рис. 2.2 – Приклад 3D тріангуляції

Щоб зробити цей простий розрахунок, GPS-приймач повинен знати дві речі:

- Розташування принаймні трьох супутників над вами
- Відстань між вами та кожним із цих супутників

Приймач GPS визначає обидві ці речі, аналізуючи високочастотні радіосигнали малої потужності від супутників GPS. Кращі пристрої мають кілька приймачів, тому вони можуть приймати сигнали з кількох супутників одночасно.

Радіохвилі є електромагнітною енергією, тобто вони рухаються зі швидкістю світла (300 000 км в секунду у вакуумі). Приймач може визначити, як далеко пройшов сигнал за часом, скільки часу знадобилося, щоб сигнал прийшов.

Як ми вже знаємо, що приймач GPS обчислює відстань до супутників GPS, визначаючи час подорожі сигналу від супутника до приймача. Як виявилось, це досить складний процес.

У певний час (скажімо, опівночі) супутник починає передавати довгий цифровий шаблон, який називається псевдовипадковим кодом. Приймач починає працювати за тим же цифровим шаблоном також рівно опівночі. Коли сигнал супутника досягає приймача, його передача шаблону буде трохи відставати від відтворення шаблону приймачем.

Тривалість затримки дорівнює часу проходження сигналу. Приймач множить цей час на швидкість світла, щоб визначити, як далеко пройшов сигнал. Припускаючи, що сигнал проходить по прямій, це відстань від приймача до супутника.

Для цього вимірювання і приймачу, і супутнику потрібні годинники, які можна синхронізувати до нано-секунд. Щоб створити систему позиціонування супутника з використанням тільки синхронізованих годин, вам потрібно мати атомні годинники не тільки на всіх супутниках, а й у самому приймачі. Але атомні годинники коштують десь від 50 000 до 100 000 доларів, що робить їх занадто дорогими для повсякденного використання.

Глобальна система позиціонування має розумне, ефективне рішення цієї проблеми. Кожен супутник містить дорогий атомний годинник, але сам приймач використовує звичайний кварцовий годинник, який він постійно скидає. Коротше кажучи, приймач дивиться на входні сигнали від чотирьох або більше супутників і вимірює власну неточність. Іншими словами, є лише одне значення для «поточного часу», яке може використовувати приймач. Правильне значення часу призведе до вирівнювання всіх сигналів, які отримує приймач, в одній точці простору. Це значення часу є значенням часу, яке зберігають атомні годинники на всіх супутниках. Таким чином, приймач встановлює свій годинник на це

значення часу, а потім має таке ж значення часу, яке мають усі атомні годинники на всіх супутниках. GPS-приймач отримує точність атомного годинника «безкоштовно».

Коли вимірюється відстань до чотирьох розташованих супутників, то можна намалювати чотири сфери, які всі перетинаються в одній точці. Три сфери будуть перетинатися, навіть якщо ваші числа далекі, але чотири сфери не перетнуться в одній точці, якщо ви виміряли неправильно. Оскільки приймач виконує всі вимірювання відстані за допомогою власного вбудованого годинника, усі відстані будуть пропорційно невірними.

Приймач може легко розрахувати необхідне налаштування, яке призведе до перетину чотирьох сфер в одній точці. На основі цього він скидає свій годинник, щоб синхронізуватися з атомним годинником супутника. Приймач робить це постійно, коли він увімкнений, що означає, що він майже такий же точний, як дорогі атомні годинники на супутниках.

Для того, щоб інформація про відстань була корисною, приймач також повинен знати, де насправді знаходяться супутники. Це не особливо складно, оскільки супутники рухаються на дуже високих і передбачуваних орбітах. Приймач GPS просто зберігає альманах, який повідомляє йому, де кожен супутник має бути в будь-який момент часу. Такі речі, як притягання Місяця та сонця, дуже незначно змінюють орбіти супутників, але Міністерство оборони постійно контролює їх точне розташування та передає будь-які налаштування всім приймачам GPS як частину сигналів супутників.

## **2.2. Технологія SD-Card**

Багато ноутбуків і мобільних пристроїв, які ви можете придбати сьогодні, мають вбудовані слоти для карт пам'яті Secure Digital, які зазвичай називаються просто картками SD. SD-карта – це пристрій зберігання даних, який має багато корисних функцій залежно від того, як і де вона використовується. Нижче наведено кілька поширених способів використання карт SD:

- Ви можете додати SD-карту до невеликого пристрою, наприклад мобільного телефону, щоб розширити простір для зберігання мелодії дзвінка, текстів, програм, музики та інших даних.
- Коли одна SD-карта у вашій камері заповнюється фотографіями та відео, ви можете замінити її на порожню SD-карту замість того, щоб видаляти будь-які файли.
- Ви можете зберігати файли на карті SD, щоб передавати їх між пристроями або передавати комусь іншому.

SD-карта — це твердо-тілий пристрій. Це означає, що він не має рухомих частин, щоб функціонувати. Це значне покращення порівняно зі старими портативними пристроями зберігання даних. Дискети, наприклад, мали крихкі диски, всередині яких крутилися з високою швидкістю і підлягали втраті даних, якщо їх розміщувати поблизу магнітних полів. Компакт-диски та DVD-диски більші та менш довговічні, а їх можливості зберігання швидко погіршуються через велику кількість перезаписів. Компоненти SD-карти є частиною її схеми, а її упаковка невелика та міцна в порівнянні з цими попередниками.

SD-карти відповідають стандартам, розробленим і підтримуваним Асоціацією Secure Digital. Panasonic, SanDisk і Toshiba створили асоціацію SD у січні 2000 року з метою розробки та просування стандартів для зберігання флеш-пам'яті. Технологія SanDisk, яка вже є світовим лідером у продуктах флеш-пам'яті, стала шаблоном для нових стандартів.

На перший погляд, SD-карта – це невеликий плоский прямокутний об'єкт з виїмкою вздовж одного краю та мідними проводами, які називаються контактами, вбудованими з одного боку вздовж іншого краю. Але є кілька різних форм-факторів. Ось фізичні розміри, які ви побачите для карт SD:

- 32 x 24 міліметри, товщина 2,1 міліметра - це найпоширеніший розмір, який ви побачите. Цей розмір має дев'ять контактів і має перемикач захисту від запису з одного боку для перемикання між станами «лише читання» та «читання-запис».

- 20 x 21,5 міліметра, товщина 1,4 міліметра - це версія «міні» з 11 штифтами, і вона зустрічається рідше, ніж два інших розміри.
- 15 x 11 міліметрів, товщиною 1 міліметр - це "мікро" версія з вісьмома шпильками. Оскільки він такий маленький, цей розмір став популярним у мобільних пристроях, таких як смартфони.

Стандартизований не лише фізичний склад картки. SD-карти також бувають різної ємності. Нижче наведено формати ємності для SD-карт у порядку від найстарішого до новітнього:

- SD – стандартний формат до 2 ГБ на карту, доступний у всіх трьох розмірах, до 10 доларів США
- SDHC - формат високої ємності з 4-32 ГБ на карту, доступний у всіх трьох розмірах, від 10 до 100 доларів США
- SDXC - формат розширеної ємності з 32 ГБ до 2 ТБ на карту, доступний лише у великих розмірах і розмірах мікро, починаючи з 80 доларів США

Це три формати ємності, кожен у двох або трьох розмірах. Щоб зчитувати та записувати дані на карту, вам знадобиться обладнання для зчитування карт, яке може обробляти як коефіцієнт розміру, так і вказаний клас ємності. Крім того, якщо ви використовуєте картку міні або мікро розміру, але маєте зчитувач великого розміру, ви можете вставити карту меншого розміру в спеціальний адаптер у формі картки більшого розміру. Деякі виробники карт SD навіть продають карти microSDHC разом із адаптерами розміром SDHC, щоб полегшити переміщення карт між пристроями.

Ще один показник, який можна побачити на SD-картці, - це її швидкість. Швидкість не дуже важлива, якщо ви просто зберігаєте файли або фотографуєте. Однак швидкість дуже важлива, коли справа доходить до зйомки та відтворення відео, особливо високої чіткості (HD). Якщо у вас швидкісна SD-карта, ви можете записувати відео кращої якості з більш плавним відтворенням.

Клас швидкості поширюється на всі карти SD. Ви побачите клас швидкості для карти SD поруч із її логотипом SD у вигляді числа всередині круглої форми «С». Нижче наведено класи швидкості, визначені асоціацією SD та картами, а

також якості запису та відтворення відео, які зазвичай асоціюються з цими класами:

- 2 - Стандартна чіткість (SD), максимальна швидкість 12,5 МБ в секунду
- 6 або 4 - HD (720p, 1080i, 1080p), максимальна швидкість 25 МБ на секунду
- 10 - Full HD (1080p), максимальна швидкість 25 МБ в секунду
- Ультра-високошвидкісний (UHS) клас швидкості доступний на SD-картах, які мають інтерфейс шини UHS-I. Це означає, що карта SD має схему, призначену для читання та запису в пам'ять зі швидкістю до 312 МБ на секунду. Упаковка картки SD UHS може мати клас швидкості плюс додатковий клас, записаний у вигляді числа у формі «U». Однак на самій картці клас швидкості UHS позначений римською цифрою «I» праворуч від логотипу SD.

Що робить карту пам'яті Secure Digital безпечною? Усі карти SD мають захист авторських прав на стандартні типи даних SD. Це дає змогу комерційним дистриб'юторам розміщувати музику чи інші носії на пристрої, щоб він був захищений від незаконного копіювання. Технологія, яка лежить в основі цієї функції картки SD, називається Content Protection for Recordable Media (CPRM).

CPRM був розроблений 4C Entity, LCC, організацією, яка займається ліцензуванням захисту авторських прав для IBM, Intel і співзасновників SD Panasonic і Toshiba. 4C Entity також розробила відповідну технологію Content Protection for Pre-Recorded Media (CPPM), яка відноситься до формату DVD-Audio для музики. CPRM і CPPM зазвичай асоціюються з форматами Flash і DVD.

CPRM не застосовується автоматично лише тому, що ви використовуєте SD-карту. Щоб дані були захищені за допомогою CPRM, пристрій, який записує дані, має спочатку створити ідентифікатор медіа та блок медіа-ключа (МКВ) на картці SD. Ідентифікатор медіа та МКВ записуються на фізичну частину карти SD, відому як захищена зона. Дані, що зберігаються в захищеній зоні, недоступні через файлову систему SD-карти, але будь-який пристрій із підтримкою CPRM, який читає SD-карту, може використовувати їх для операцій із захисту вмісту.

Щоб захистити дані за допомогою CPRM, пристрій використовує ідентифікатор медіа та МКВ для кодування даних під час запису на карту SD. Щоб прочитати дані, пристрій із підтримкою CPRM отримує доступ до Media Identify і МКВ і використовує їх для розшифрування даних під час зчитування з SD-карти. Тому, навіть якщо ви скопіюєте захищені CPRM файли з файлової системи SD-карти, ви не зможете прочитати їх за межами цієї SD-карти.

На додаток до CPRM є ще одна функція безпеки, яку ви помітите на 32 x 24-міліметрових картах SD (SD, SDHC і SHXC): блокування захисту від запису. Замок — це крихітний повзунок вздовж однієї сторони SD-карти. У розблокованому положенні ви можете читати та записувати на SD-карту, як з USB-накопичувачем або будь-яким іншим пристроєм для читання-запису. У заблокованому положенні SD-карта не дозволить змінювати файлову систему, що робить її пристроєм лише для читання. Ви можете використовувати цю функцію, якщо ви стурбовані випадковим перезаписом даних із камери, наприклад, під час читання SD-карти на ноутбуком чи настільним комп'ютером.

### **2.3. Використання технологій в розроблюваній автоматизованій системі**

Автоматизована система вимірювання напруження електричного поля атмосфери на місцевості, має в автоматичному режимі відслідковувати переміщення датчика, та зберігати дані про вимір в кожній точці, та координати цієї точки. Саме для фіксації переміщень та отримання координат місцезнаходження датчика і буде відповідати GPS-модуль, а за зберігання даних буде відповідати SD-Card, так як це пам'ять довготривалого зберігання даних,



має хорошу швидкість обміну даних, а також дані можна зручно передати на ПК, чи ноутбук для подальшої обробки отриманих значень.

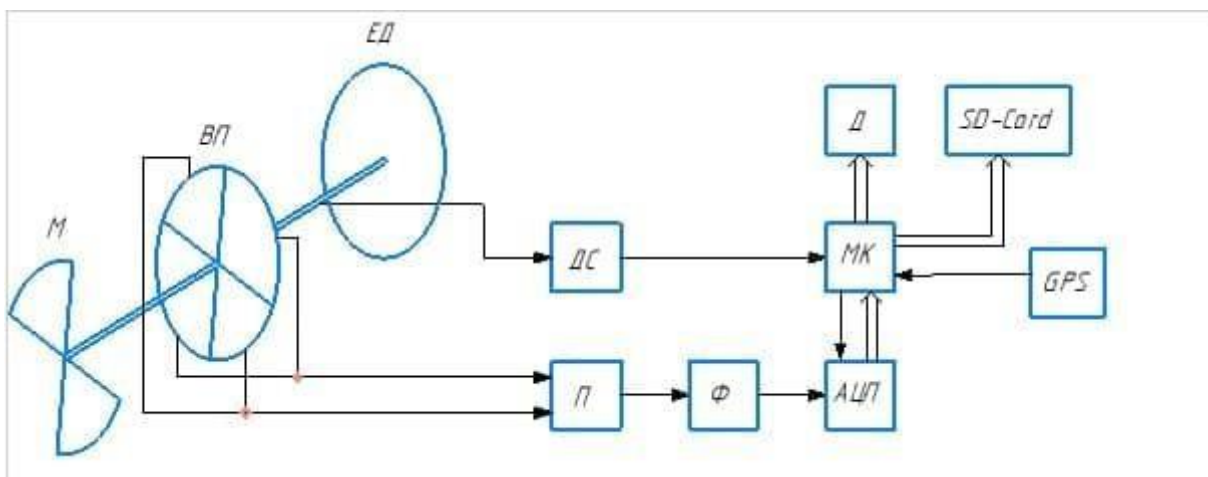


Рис. 2.3 – Структурна схема системи

На Рис. 2.3 показано структурну схему системи для автоматизованого вимірювання напруженості електричного поля в атмосфері. Саме на ній й показано що GPS й SD-Card напряму підключаються до мікроконтролера, та напряму взаємодіють з ним.

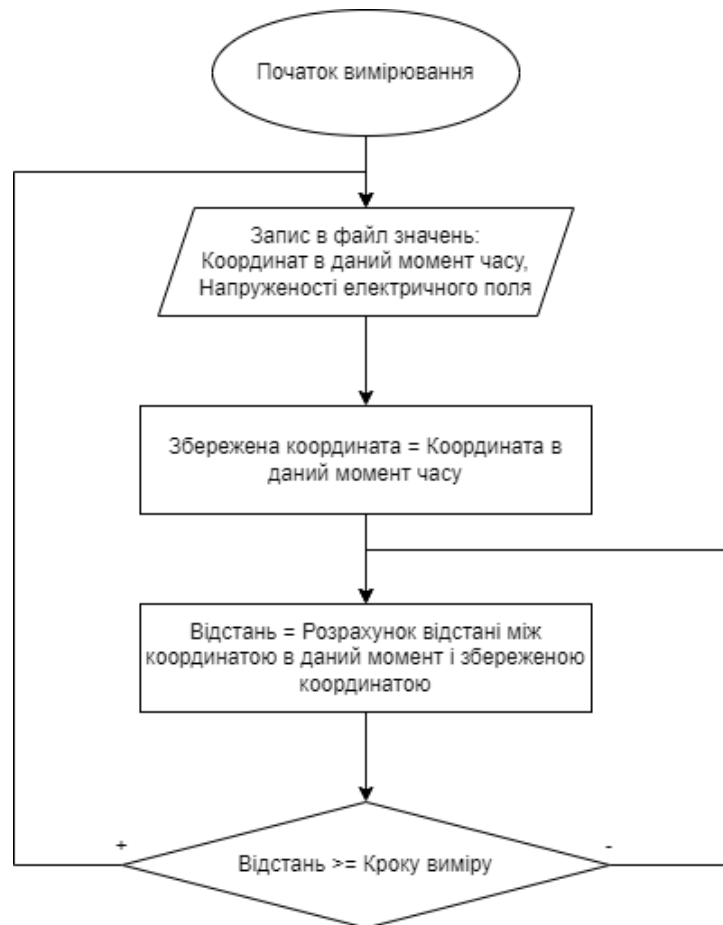


Рис. 2.4 – Блок-схема виконання автоматизованого вимірювання напруженості електричного поля на місцевості

На Рис. 2.4 представлена блок-схема, яка демонструє порядок виконання автоматизованого вимірювання напруженості електричного поля на місцевості.

Одразу після включення приладу система починає виконувати виміри. Спочатку вона записує координату отриману через GPS відразу після включення, і значення напруженості електричного поля атмосфери в файл. Далі вона зберігає цю координату де був зроблений вимір і записаний в файл. Після цього система буде постійно отримувати дані з GPS – модуля про координату в якій знаходиться наша система. На базі отриманої координати буде вираховуватись відстань до збереженої координати, і після цього порівнюватись з заданим кроком виміру. Якщо відстань менше за крок виміру, то система буде знов брати координату в даний момент, і вираховувати відстань. Якщо ж відстань буде рівна або більша кроку, то ми записуємо це значення в файл, і обновляємо змінну зі збереженою координатою. Так буде продовжуватись поки буде ввімкнутий

прилад. Як тільки прилад вимкнеться, то припиниться вимірювання, а на SD-картці отримаємо файл з координатами, та значеннями напруженості електричного поля атмосфери в цих координати.

## 2.4. Структурний розрахунок приладу

Ключовими параметрами нашого приладу є швидкодія, точність, маленький розмір, енергоефективність, та маленька ціна. Виходячи з цього був проведений структурний розрахунок приладу, для отримання найоптимальніших значень цих параметрів.

Виходячи з ключових параметрів був підібраний мікроконтролер STM32F103C8T6, цей мікроконтролер має хороші розрахункові потужності, яких вистачає для виконання задач нашим пристроєм. На це вказують такі параметри як частота мікроконтролера 72МГц, оперативна пам'ять 64 кБ, має два 12-бітних АЦП з мінімальною частотою дискретизації 0.6МГц, а максимальною 14МГц, а також він є досить економічним по енерговитратам, та невеликих розмірів, вартість даного мікроконтролера близько 100грн, що повністю задовольняє нашим вимогам.

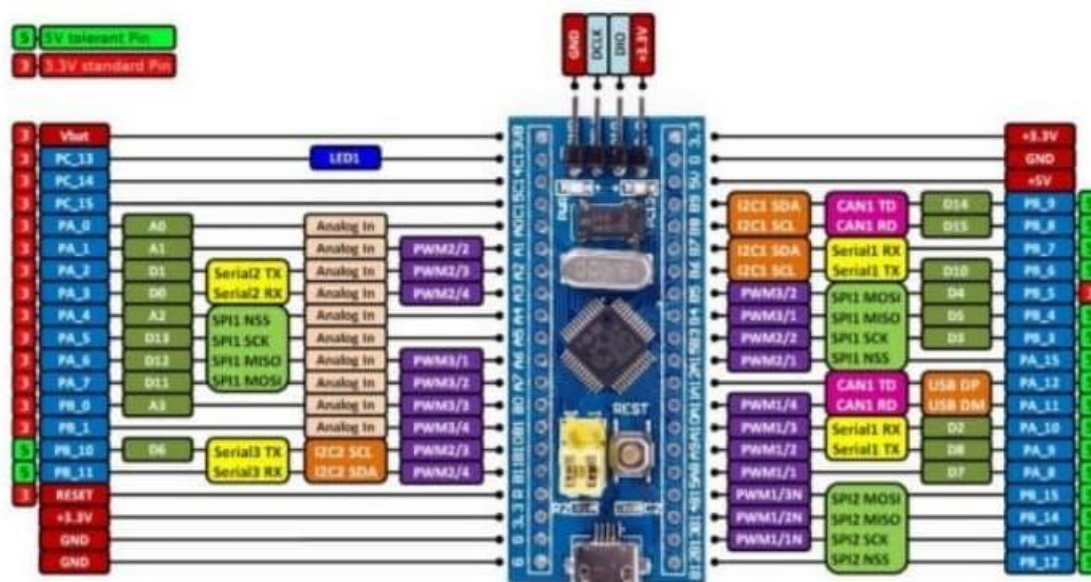


Рис. 2.5 – STM32F103C8T6

Для реєстрації позиціонування системи в просторі використовується GPS-модуль NEO-7M. Даний GPS – модуль було вибрано з міркувань достатньої

точності, його точність складає – 2.5 метрів. Він є енергоефективним, та має інтерфейс підключення UART, що дає нам змогу підключити його до мікроконтролера STM32F103C8T6, таким чином він повністю відповідає нашим вимогам.

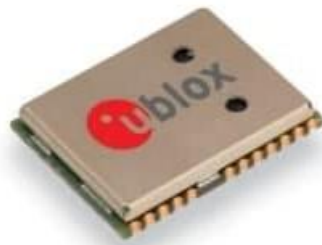


Рис. 2.6 – GPS – модуль NEO-7M

Далі було підібрано двигун, двигун потрібен був з частотою обертання більше 1КГц, та має бути малих розмірів, для збереження компактності приладу і недорогий. За цими параметрами було підібрано двигун RF-300CA-09550. Його частота обертання 2.7КГц, коштує 27.5 грн, а його розміри приведені на Рис. 2.7. Що демонструють компактність датчика.

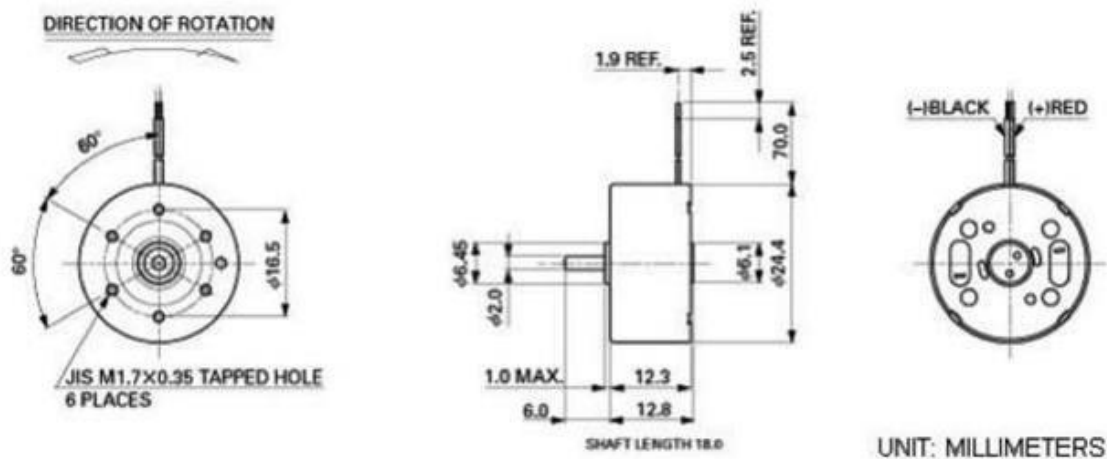


Рис. 2.7 – Розміри двигуна

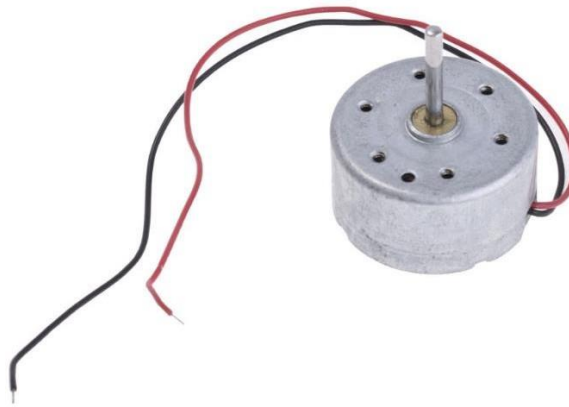


Рис. 2.8 – Зовнішній вигляд двигуна

Також для нашої системи був підібраний енкодер, це потрібно для визначення повороту валу на четверть, так як це свідчить про перехід від однієї пари екрануючих пластин, до другої. Також за допомогою енкодера можна підвищити точність вимірювання. Розглянемо яким способом це можна реалізувати, так як енкодер буде реєструвати поворот валу на четверть, то вимірявши час за який він це буде робити, ми зможемо визначити миттєве значення періоду обертання, і підставити значення в формулу по якій система розраховує напруженість поля атмосфери. Таким чином ми підняли точність, так як до цього це значення було константою, але по факту є чинники, які можуть впливати на зміну періоду обертання, такі наприклад як, зменшення заряду акумулятора, при зменшенні заряду, період обертання збільшиться, ми будемо вимірювати одні дані, і при їх обробці в напруженість, буде підставлятись не правильне статичне значення періоду. А в моєму варіанті буде підставлятись миттєве вимірне значення, що майже повністю усуне цю похибку.

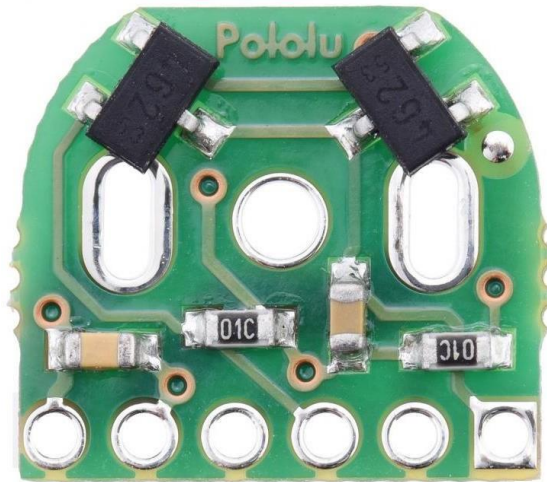


Рис. 2.9 – Магнітний енкодер Pololu

На даному датчику встановленні два датчика Холла, кут між якими 90 градусів. Але він комплектується шести полюсним магнітом, що дозволяє визначити 12 положень валу, що занадто багато для нашого приладу, тому даний магніт можна замінити на магніт з такими самими характеристиками, але двох полюсний, що в комплексі з двома датчиками Холла, дасть нам можливість визначати чотири положення валу.

Також в рамках даного розділу, було зроблено покращення алгоритму вимірювання. Розглянемо, як само наш алгоритм покращився.

Пластини поперемінно екранують, та експонують електричне поле між собою. Через це на виході ми отримуємо змінний струм. Для обчислення нашою системою значення напруженості нам потрібно знайти амплітуду сигналу, між максимальним та мінімальним значенням. Для знаходження цієї амплітуди можна використовувати метод з перетворенням Фур'є, або метод найменших квадратів, але обидва методи є дуже складними, тому дають велике обчислювальне навантаження, а також збільшують час виміру, тому пропонується альтернативний алгоритм виміру.

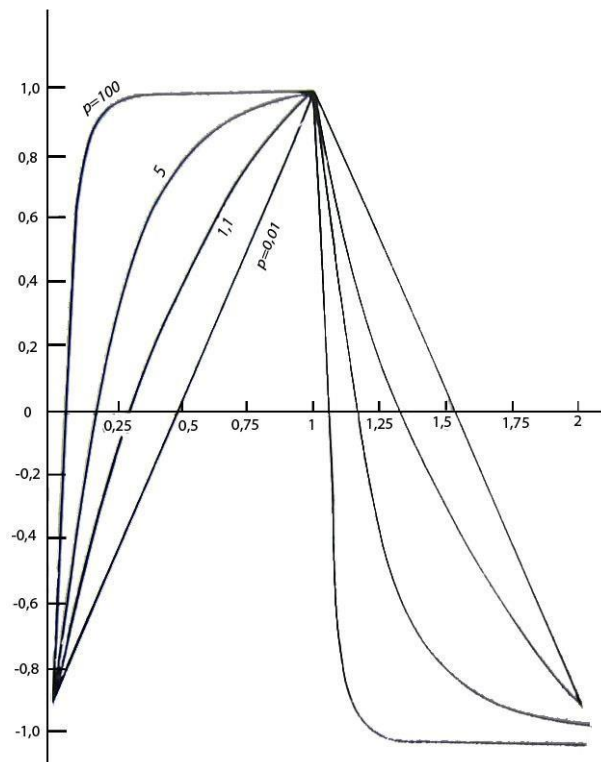


Рис. 2.10 – Форма сигналу для різних значень параметру Р для  $m = 0,2$

На даному графіку можна побачити, що при  $p = 100$  сигнал довго тримається рівним в максимумі, та в мінімум, тому ми можемо зачекати третину періоду, і за наступну третину зробити фіксацію трьох значень, вони всі будуть зроблені або в максимумі, або в мінімумі, а три їх робиться для подальшого усереднення, так як на сигнал може бути з шумом. Таким чином ми можемо легко виміряти максимальне і мінімальне значення сигналу, а отже і обрахувати амплітуду. Тепер розглянемо як зробити так, щоб  $p$  було рівним 100. З формули (2.1) бачимо що  $p$  залежить від періоду, і ємності.

$$p = \frac{T}{2 * R * C_n} \quad (2.1)$$

А ємність в свою чергу залежить від площі пластин, саме нашої пластини яка крутиться, і екрануючих.

## 2.5. Висновки до розділу

В даному розділі було розроблено систему за допомогою якої можна автоматизувати процес вимірювання напруженості електричного поля атмосфери на місцевості. В ході розробки було встановлено, що для

автоматизації даних вимірювань, потребується використання таких технологій, як GPS, та SD-card, в розділі наданого огляд даних технологій, а також опис їх роботи. Складено алгоритм роботи автоматизованої системи разом з даними технологіями. Також в рамках даного розділу було проведено аналіз обраного нами методу вимірювання, та розглянуто шляхи покращення.



### **3. Опис роботи програми**

В рамках моєї магістерської роботи також було створено спеціальне програмне забезпечення, для обробки даних вимірювання та інформативної подачі результату. Головною задачею даного ПЗ, є перетворення даних з файлу отриманого з автоматизованої системи, в карту місцевості на яку додатковим шаром буде нанесено градієнт розподілу напруженості електростатичного поля атмосфери на цій місцевості.

Для написання програмного забезпечення для нашої системи використовується мова програмування C#.

C# - мова високого рівня побудована на об'єктно – орієнтованій парадигмі. Найчастіше використовується з платформою .NET Framework. Саме на цій платформі й була написана наша програма. Мова програмування C# належить корпорації Microsoft.

Синтаксис C# близький до C++ і Java. Мова має строгу статичну типізацію, підтримує поліморфізм, багатопоточність, перевантаження операторів, вказівники на функції-члени класів, асинхронне програмування атрибути, події, властивості, коментарі у форматі XML. Переїнявши багато від своїх попередників — мов C++, Smalltalk, Object Pascal, і Модула — C#, спираючись на їхню практику використання, виключає деякі моделі, що показали себе як проблематичні при розробці програмних систем, такі як, множинне наслідування, тому C# на відміну від C++ його не підтримує.

В ході написання даної програми використовувався фреймворк Windows Forms. Це спеціальний фреймворк для створення десктопних програм з графічним інтерфейсом.

#### **3.1. Використані бібліотеки**

При створенні даного ПЗ були використанні наступні бібліотеки:

- System
- GMap.NET

- GMap.NET.WindowsForms

Бібліотека System – потрібна для використання загальних функцій мови C#, наприклад таких функції запису, та читання з файлу, роботи з потоками та асинхроностями, а також спеціальним типом масивів List.

Бібліотека GMap.Net – це спеціальна бібліотека для роботи з картами від різних постачальників, а також для повного функціоналу , та зручної роботи з картами. Зручна робота полягає в включених в бібліотеку спеціальних класів, які підтримують створення точок з координатами в форматі широти, довготи.

Бібліотека GMap.NET.WindowsForms – це бібліотека для сполучення GMap.NET та фреймворку Windows Forms, за допомогою спеціальних інструментів. Наприклад вона передбачує додавання в базовий набір інструментів Windows Forms, спеціального вікна GMapControl, на який можна виводити зображення карти, від постачальників.

### 3.2. Код програми

Перейдемо до розгляду самої програми, та її коду. Перейдемо до розташування графічних елементів нашої програми.

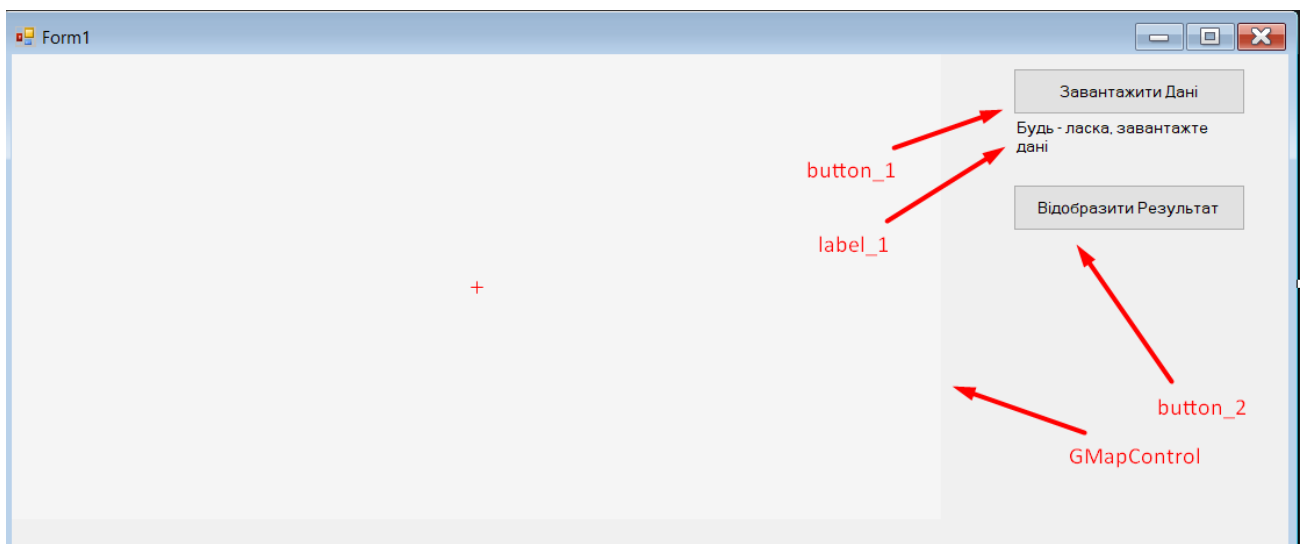


Рис. 3.1 – Графічне вікно програми

На Рис. 3.1 – відображена графічне вікно мого ПЗ, з розташованими в ній елементами:

- GMapControl – це графічний елемент, з бібліотеки GMap.NET.Windows Forms, на якому буде відображатись наша карта, та додактовий шар;
- Button\_1 – це кнопка яка викликатиме діалогове вікно, для вибору файла з якого ми будемо вести побудову додаткового шару з градієнтом розподілу напруженості електростатичного поля атмосфери на ній;
- Label\_1 – це поле, в якому буде відображатись поточний статус програми, або що потрібно зробити;
- Button\_2 – це кнопка, яка буде використовуватись для відображення результату.

Тепер розглянемо сам код, і почнемо з коду ініціалізації самої форми.

```

using System;
using System.IO;
using System.Threading.Tasks;
using System.Windows.Forms;

using GMap.NET;
using GMap.NET.WindowsForms;

namespace StrengthElectrostaticFieldMap
{
    public partial class Form1 : Form
    {
        StrengthMapOverlay sMapOverlay;
        string Data;

        public Form1()
        {
            InitializeComponent();
        }

        private async void button1_Click(object sender, EventArgs e)
        {
            // ...
        }

        private void gMap_Load(object sender, EventArgs e)
        {
            // ...
        }

        private void button2_Click(object sender, EventArgs e)
        {
            // ...
        }
    }
}

```

Рис. 3.2 – Клас Form

На Рис. 3.2 – зображено код, який виконується при ініціалізації форми, в самому верху якого ми бачимо підключення бібліотек які ми використовуємо, далі визначається простір імен в якому ми працюємо, та код самого класу Form1 в якому міститься 4 метода. Розглянемо кожен метод детальніше:

```

public Form1()
{
    InitializeComponent();
    openFileDialog1.Filter = "Text files(*.txt)|*.txt";
}

```

Рис. 3.3 – Метод - конструктор

На Рис. 3.3 – зображене тіло методу Form1, так званого ‘метода – конструктора’, тобто він виконується кожен раз при створенні об’єкта даного класа. В цьому методі викликається функція InitializeComponent, яка по черзі ініціалізує кожен окремий компонент в формі, що знаходиться в графічному інтерфейсі. В наступному рядку вказується, що через діалогове вікно OpenFileDialog1 можна відкрити файл тільки з розширенням txt. Саме такий файл який буде надходити нам з нашої системи.

```
private async void button1_Click(object sender, EventArgs e)
{
    if (openFileDialog1.ShowDialog() == DialogResult.Cancel)
        return;
    string filename = openFileDialog1.FileName;
    using (var sr = new StreamReader(filename)) {
        Data = sr.ReadToEnd();
    }
    label1.Text = "Очікуйте йде обробка даних";
    await Task.Run(() => sMapOverlay = new StrengthMapOverlay("Karpaty", gMap, Data));
    button2.Enabled = true;
    label1.Text = "Обробка закінчена.\nНатисніть відобразити результат.";
}
```

Рис. 3.4 – Метод button1\_Click

Метод button1\_Click зображений на Рис. 3.4, це метод обробник натискання на кнопку з іменем button1, це кнопка з надписом ‘Завантажити Дані’. Як бачимо даний метод має атрибут async, він означає, що операції в даному методі можуть бути запущені в асинхронному режимі.

В даному методі обробнику вказано що при натисканні на кнопку, буде відкрито діалогове вікно, для вибору файла, шлях та назва файла, буде збережена в змінну filename. Після цього відкриється безпечний потік, в якому ми читаємо всі дані з файла в змінну Data. Далі змінюємо текст, в текстовому вікні на такий що в нас починається обробка даних. Наступний рядок це створення додаткового шару для карти на якому й буде міститись наш градієнт розподілу напруженості, дана операція спеціально запускається з атрибутом await, і командою Task.Run() вона означає, що дана операція буде виконуватись повністю в асинхронному режимі.

В параметрах для створення даного об’єкта в якості параметрів ми передаємо:

- Назву шара;
- Назва об'єкта графічного елемента на якому в нас відображається карта;
- Дані які ми зчитали з файла.

В ході створення нашого об'єкта, дані зчитані з файла обробляться і перетворюються на зображення.

Після створення даного об'єкту, робимо активною кнопку button2, це кнопку з надписом 'Відобразити дані', так як дані вже були оброблені і ми можемо відобразити їх на карті.

```
private void gMap_Load(object sender, EventArgs e)
{
    gMap.DragButton = MouseButtons.Left;
    gMap.MaxZoom = 18;
    gMap.MinZoom = 2;
    gMap.Zoom = 13;
    gMap.MapProvider = GMap.NET.MapProviders.GMapProviders.GoogleMap;
    GMaps.Instance.Mode = AccessMode.ServerOnly;
    PointLatLng PositionCenter = new PointLatLng(48.5, 23.5);
    gMap.Position = PositionCenter;
}
```

Рис. 3.5 – Метод gMap\_Load

Після методу button1\_Click знаходиться метод gMap\_Load, це метод для ініціалізації графічного елемента на якому буде відображатись карта, як можна побачити в ньому задані параметри для цього елемента, розглянемо кожен з них:

- gMap.DragButton = MouseButtons.Left – в даному параметрі вказується якою кнопкою буде виконуватись перетаскування карти;
- gMap.MaxZoom = 18 – параметр в якому вказується максимальний зум для карти (Можливий діапазон зума від 1 до 18);
- gMap.MinZoom = 2 - параметр в якому вказується мінімальний зум для карти;
- gMap.Zoom = 13 параметр, який визначає який буде зум при відкритті програми
- gMap.MapProvider = GMap.NET.MapProviders.GMapProviders.GoogleMap – параметр в якому вказується карти від якого постачальник будуть використовуватись(В даному випадку від Google);

- `GMap.Instance.Mode = AccessMode.ServerOnly` – параметр в якому визначається в якому режимі будуть працювати карти(Онлайн режим, режим кеша, або змішаний режим);
- `gMap.Position = PositionCenter` – параметр в якому вказується на якій позиції карта має бути відображена при старті програми, їй передається об'єкт спеціального класу `PointLatLng`, що входить в бібліотеку `GMap.NET`, і слугує для збереження точки, форматі координат широти та довготи.

```
private void button2_Click(object sender, EventArgs e)
{
    gMap.Overlays.Add(sMapOverlay);
    label1.Text = "Дані виведені на карту";
    button2.Enabled=false;
}
```

Рис. 3.6 – Метод `button2_Click`

Метод `button2_Click` зображений на Рис. 3.6, це метод обробник натискання на кнопку `button2`. Тут ми бачимо що до графічного елемента на якому відображається карта, додається шар, на якому знаходиться градієнт розподілу напруженості. Також оновлюється текст в текстовому полі, на відповідний, і кнопка `button2` стає не активна.

Тепер розглянемо створений мною клас `StrengthMapOverlay`. Загальний вигляд коду класу представлено на Рис. 3.7.

```
using System.Drawing;
using GMap.NET;
using GMap.NET.WindowsForms;

namespace StrengthElectrostaticFieldMap
{
    3 references
    public class StrengthMapOverlay : GMapOverlay
    {
        GMapControl mapControl;
        ZoomCache[] cache;
        Point startLocalPoint;
        int presentZoom;
        1 reference
        public StrengthMapOverlay(string id, GMapControl mapControl, string point) {...}
        0 references
        public override void OnRender(Graphics g) {...}
    }
}
```

Рис. 3.7 – Клас `StrengthMapOverlay`

Як можна побачити з Рис. 3.7, мій клас `StrengthMapOverlay` наслідується з класу `GMapOverlay`, що являється спеціальним класом з бібліотеки `GMap.NET.WindowsForms`, для створення додаткових шарів. Але в бібліотеці

GMap.NET.WindowsForms, передбачено рисування на додаткових шарах, тільки маркерів, маршрутів, та полігонів що не є підходящим для нашої задачі, тому я і створив свій власний шар.

Клас StrengthMapOverlay має визначені базові поля:

- GMapControl mapControl – для передачі об'єкта графічного елемента на якому малюється карта;
- ZoomCache[] cache – для збереження масиву наших картинок та інформації, по градієнту розподілу напруженості, для кожного зума.
- Point startLocalPoint – для зберігання координат точок з якого буде починатись побудова зображення;
- int presentZoom – для зберігання даних про зум в даний момент часу.

Також в класі присутні два методи: StrengthMapOverlay, та onRender.

Розглянемо кожен з них детальніше.

```
public StrengthMapOverlay(string id, GMapControl mapControl, string point)
{
    Id = id;
    this.mapControl = mapControl;
    cache = Drawer.GetZoom(point, mapControl.MinZoom, mapControl.MaxZoom);
    startLocalPoint = new Point((int)(cache[(int)mapControl.Zoom].startPoint.X
        + mapControl.PositionPixel.X), (int)(cache[(int)mapControl.Zoom].startPoint.Y - mapControl.PositionPixel.Y));
    presentZoom = (int)mapControl.Zoom;
}
```

Рис. 3.8 – Метод StrengthMapOverlay

Метод StrengthMapOverlay є так званий методом конструктором класу. Він викликається при створенні об'єкта даного класу через new.

Розглянемо, які дії відбуваються в цьому методі. В першу чергу присвоюється Id на той що був переданий при створенні об'єкту. Хоч ми й не оголошували поле Id в данному класі, але воно було взяте з класу з якого ми наслідувались. Далі присвоюється переданий при створенні об'єкт графічного елемента, на якому відображається карта, відповідному полю об'єкта.

Наступним кроком є присвоювання відповідному полю об'єкта cache, результату відпрацювання функції Drawer.GetZoom(), який повертає масив об'єктів який містить картинку градієнта розподілу напруженості

електростатичного поля атмосфери, для кожного з зумів, та додаткових параметрів наприклад координати точки прикріплення нашого зображення, до карти. Далі присвоюється об'єкту `startLocalPoint`, розрахована по виразу, точка з координатами для прикріплення зображення відносно того зображення, що зараз знаходиться на графічному елемента відображення карти. Ну і в змінну `presentZoom` присвоюється значення зуму в теперішній час.

```
public override void OnRender(Graphics g)
{
    if (presentZoom != mapControl.Zoom)
    {
        startLocalPoint = new Point((int)(cache[(int)mapControl.Zoom].startPoint.X - mapControl.PositionPixel.X),
            (int)(cache[(int)mapControl.Zoom].startPoint.Y - mapControl.PositionPixel.Y));
        presentZoom = (int)mapControl.Zoom;
    }
    g.DrawImage(cache[(int)mapControl.Zoom].picture, startLocalPoint);
}
```

Рис. 3.9 – Метод `onRender`

На Рис. 3.9 зображено код методу `onRender`. Атрибут `override` – вказує на те, що цей метод буде перевизначено, це означає, що такий самий метод є в класі від якого ми наслідували наш клас, і тепер буде подана друга його реалізація, саме для мого об'єкта.

В самому методі ми кожен раз перевіряємо запам'ятований зум, з тим що є зараз, і якщо він відрізняється, це означає що відбулась зміна приближення карти, а отже потрібно і перерахувати точку прив'язки до карти, для нового зума, що і відбувається в наступних рядках, і далі запам'ятовуємо зум для якого було зроблено розрахунок, для його подальшого зрівняння. Ну і останній рядок ми малюємо зображення градієнта розподілу напруженості, беручи його з масиву об'єктів для кожного зума, та розраховану локальну точку прив'язки до карти, від якої і буде починатись малювання зображення.

Далі розглянемо клас `Drawer` створений мною для створення зображень для всіх приближень карти, що в нас можуть бути. Загальна структура коду даного класу зображена на Рис. 3.10



```

1 using System.Drawing;
2 using System.Linq;
3
4 using GMap.NET;
5 using GMap.NET.Projections;
6
7 namespace StrengthElectrostaticFieldMap
8 {
9     1 reference
10    public static class Drawer
11    {
12        old
13        1 reference
41    public static ZoomCache[] GetZoom(string Data, int minZoom, int maxZoom) ...
14        1 reference
64    public static StrengthLatLng[,] GetMatrixArray(string Data) ...
15        1 reference
81    public static StrengthLatLng[] GetLineArray(string Data) ...
16        1 reference
96    public static int GetBasicZoom(StrengthLatLng[,] MatrixData, MercatorProjection projection) ...
17        1 reference
112    public static StrengthPoint[,] GetPixelArray(StrengthLatLng[,] MatrixData, int zoom, MercatorProjection projection) ...
18        1 reference
147    public static Bitmap getPictureFromPixelArray(StrengthPoint[,] pixelArray) ...
19        1 reference
162    public static Bitmap getUpperPicture(Bitmap picture, GPoint StartPoint, GPoint EndPoint) ...
171
172    }
173

```

Рис. 3.10 – Клас Drawer

Сам клас Drawer статичний і як всі його методи, це означає що ми можемо їх використовувати без створення екземпляра самого класу. Так як створення зображень для всіх шарів, є однаковою процедурою, яка майже ніяк не залежить від самого класу, і потребує певну невелику кількість параметрів, то щоб не перевантажувати сам клас шарів, було прийнято рішення винести всі методи для створення в окремий статичний клас. Розглянемо кожен метод окремо, для чого він потрібен, і як працює.

```

public static ZoomCache[] GetZoom(string Data, int minZoom, int maxZoom)
{
    MercatorProjection projection = new MercatorProjection();
    StrengthLatLng[,] MatrixDataLatLng = GetMatrixArray(Data);
    int basicZoom = GetBasicZoom(MatrixDataLatLng, projection);
    ZoomCache[] zoomCache = new ZoomCache[19];
    for (int i = basicZoom; i >= minZoom; i--)
    {
        StrengthPoint[,] pixelArray = GetPixelArray(MatrixDataLatLng, i, projection);
        GPoint FirstPoint = projection.FromLatLngToPixel(MatrixDataLatLng[MatrixDataLatLng.GetLength(0) - 1, 0].pointLatLng, i);
        Bitmap picture = getPictureFromPixelArray(pixelArray);
        zoomCache[i] = new ZoomCache(FirstPoint, picture, pixelArray);
    }
    for (int i = basicZoom + 1; i <= maxZoom; i++)
    {
        GPoint FirstPoint = projection.FromLatLngToPixel(MatrixDataLatLng[MatrixDataLatLng.GetLength(0) - 1, 0].pointLatLng, i);
        GPoint SecondPoint = projection.FromLatLngToPixel(MatrixDataLatLng[0, MatrixDataLatLng.GetLength(1) - 1].pointLatLng, i);
        Bitmap picture = zoomCache[i - 1].picture;
        Bitmap bitmap = getUpperPicture(picture, FirstPoint, SecondPoint);
        zoomCache[i] = new ZoomCache(FirstPoint, bitmap);
    }
    return zoomCache;
}

```

Рис. 3.11 – Метод GetZoom

Метод GetZoom() це і є метод до якого ми звертаємось для отримання масиву даних кожного зума, розглянемо його детальніше. Цей метод приймає 3 параметри, це змінна з даними, які ми зчитали з нашого файла, які поки що

представлені зараз одним рядком, наступний параметр, це мінімальний зум, який може бути на нашій карті, і 3 це максимальний зум, який може бути на нашій карті. Для зручності розглядання нашого методу розіб'ємо його 3 на блоки.

```
MercatorProjection projection = new MercatorProjection();  
StrengthLatLng[,] MatrixDataLatLng = GetMatrixArray(Data);  
int basicZoom = GetBasicZoom(MatrixDataLatLng, projection);  
ZoomCache[] zoomCache = new ZoomCache[19];
```

Рис. 3.12 – Перший блок GetZoom

Отже в першому блоці, який зображено на Рис. 3.12 ми можемо побачити, в першому рядку, створення об'єкту класу MercatorProjection, який являється класом бібліотеки GMap.NET.Projection, і потрібен нам для визначення проекції з координат точки в вигляді, широти, довготи, в вигляді координат X, та Y пікселя, на всій карті. Далі створюємо двомірний масив даних з створеного мною класу StrengthLatLng і присвоюємо йому результат роботи методу GetMatrixArray, з переданою йому змінною в якій міститься дані зчитані з файла. В цьому двомірному масиві буде зберігатись матриця, точок з координатами в вигляді широти, та довготи, і значення напруженості. Після цього створюється змінна в яку присвоюється результат роботи методу GetBasicZoom з переданим йому матрицею, та об'єкта, для створення проекцій. Дана функція поверне нам максимальне значення зума, при якому кількість точок вимірювань по кожній осі, буде більша ніж кількість пікселів по тій самій осі, тобто коли на один піксель буде припадати одне, або більше кількість вимірювань, це потрібно знати, так як побудова зображення, при кількості вимірювань один або більше на піксель, відрізняється від побудови зображень, при відсутності вимірювань на пікселі. І далі створюємо масив класу ZoomCache для зберігання даних кожного зображення градієнта розподілу для кожного значення зума з кількістю елементів 19, так як максимальний можливий зум 18, а рахунок в масивах починається з 0, і щоб можна було напряду звертатись до зума то беремо кількість елементів 19.

```

for (int i = basicZoom; i >= minZoom; i--)
{
    StrengthPoint[,] pixelArray = GetPixelArray(MatrixDataLatLng, i, projection);
    GPoint firstPoint = projection.FromLatLngToPixel(MatrixDataLatLng[MatrixDataLatLng.GetLength(0) - 1, 0].pointLatLng, i);
    Bitmap picture = getPictureFromPixelArray(pixelArray);
    zoomCache[i] = new ZoomCache(firstPoint, picture, pixelArray);
}

```

Рис. 3.13 – Другий блок GetZoom

В другому блоці методу GetZoom, зображеному на Рис. 3.13, бачимо цикл, який буде йти, від отриманого базового зума, до мінімального. Всередині циклу ми бачимо створення двомірного масиву з класу StrengthPoint, що був створений мною, і зберігає в собі точки в вигляді координат X та Y на зображенні градієнта, та значення напруженості в даній точці. Даному двомірному масиву присвоюється результат роботи функції GetPixelArray, якому передається двомірний масив з точками в вигляді широти і довготи, зум для якого потрібно побудувати зображення, та об'єкт, для створення проєкцій.

Далі визначаємо початкову точку відбудови зображення, вона ж і є точкою прив'язки нашого зображення до карти. Для цієї точки застосовуємо клас GPoint, що є спеціальним класом, бібліотеки GMap.NET, для зберігання точок, з координатами в вигляді X, та Y на глобальній карті. В цей об'єкт присвоюється результат роботи функції об'єкта проєкції, а саме функції перетворення, з точки з координатами в вигляді широти, довготи, в точку з координатами пікселя в вигляді X та Y, на глобальній карті.

Далі створюється об'єкт класу Bitmap, це спеціальний клас в мові програмування C#, об'єкти якого можуть зберігати в собі зображення. Даному об'єкту присвоюється результат роботи методу getPictureFromPixelArray з переданою йому масивом точок в вигляді координат на глобальній карті, та значень напруженості в них.

Ну і в кінці, в масив даних для зумів в елемент з номером відповідного зума, присвоюється значення нового об'єкта класу ZoomCache з переданими йому точки прив'язки, отриманого зображення, та масиву пікселей з значеннями напруженості.

```

for (int i = basicZoom + 1; i <= maxZoom; i++)
{
    GPoint FirstPoint = projection.FromLatLngToPixel(MatrixDataLatLng[MatrixDataLatLng.GetLength(0) - 1, 0].pointLatLng, i);
    GPoint SecondPoint = projection.FromLatLngToPixel(MatrixDataLatLng[0, MatrixDataLatLng.GetLength(1) - 1].pointLatLng, i);
    Bitmap bitmap = getUpperPicture(zoomCache[i - 1].picture, FirstPoint, SecondPoint);
    zoomCache[i] = new ZoomCache(FirstPoint, bitmap);
}
return zoomCache;

```

Рис. 3.14 – Третій блок GetZoom

В третьому блоці методу GetZoom зображеному на Рис. 3.14, бачимо цикл який буде створювати зображення, для зума який вище базового. Спочатку оголошується змінна класу GPoint, в яку присвоюється проекція точки лівої верхньої координати зображення, а другій змінній цього ж класу присвоюється значення проекції правої нижньої точки зображення. Після цього створюється об'єкт класу Bitmap, в який присвоюється результат роботи методу getUpperPicture з переданими йому зображенням градієнта попереднього приближення, лівої верхньої точки, та правої нижньої. І в кінці в масив даних приближень присвоюється об'єкт з даними про зображення градієнта, і передається йому точка прив'язки, вона же верхня ліва точка, і отримане зображення. І після виходу з циклу ми бачимо, повернення роботи даного методу, масиву з даними про зображення для всіх приближень, які нам потрібні.

Перейдемо до наступного методу GetMatrixArray, який зображено на Рис. 3.15

```

public static StrengthLatLng[,] GetMatrixArray(string Data)
{
    StrengthLatLng[] dataArray = GetLineArray(Data);
    var distlat = dataArray.Select(s => s.pointLatLng.Lat).Distinct().OrderBy(s => s).ToArray();
    var distlng = dataArray.Select(s => s.pointLatLng.Lng).Distinct().OrderBy(s => s).ToArray();

    StrengthLatLng[,] MatrixData = new StrengthLatLng[distlat.Length, distlng.Length];

    for (int i = 0; i < distlat.Length; i++)
    {
        for (int j = 0; j < distlng.Length; j++)
        {
            MatrixData[i, j] = dataArray.First(x => x.pointLatLng.Lat == distlat[i] && x.pointLatLng.Lng == distlng[j]);
        }
    }
    return MatrixData;
}

```

Рис. 3.15 – Метод GetMatrixArray

В методі GetMatrixArray, відбувається отримання матриці елементів з координатами в вигляді широти, довготи з отриманих даним з файлу. Спочатку ми створюємо одномірний масив значень типу StrengthLatLng. Далі визначаємо

унікальні елементи, сортуємо їх, і перетворюємо в масив елементів за допомогою так званих лямбда – виразів, приклад якого наведено на Рис. 3.16.

```
var distlat = dataArray.Select(s => s.pointLatLng.Lat).Distinct().OrderBy(s => s).ToArray();
```

Рис. 3.16 – Приклад рядка з використанням лямбда – виразів

Далі створюємо двомірний масив, з довжиною рівній кількості унікальних елементів по широті, і шириною рівній кількості елементів по довготі.

Далі створюємо два вкладених цикл, для заповнення двомірного масиву, для цього ми беремо значення широти з масиву унікальних широт, відповідного до індексу елемента який заповнюється, і так само робимо і з довготою, в кінці отримуємо заповнений двомірний масив, з відсортованими точками, і повертаємо його як результат роботи функції.

```
public static StrengthLatLng[] GetLineArray(string Data)
{
    string[] LineArray = Data.Split('\n');
    StrengthLatLng[] dataArray = new StrengthLatLng[LineArray.Length]

    for (int i = 0; i < LineArray.Length; i++)
    {
        string[] Line = LineArray[i].Split(';');
        double lat = double.Parse(Line[0]);
        double lng = double.Parse(Line[1]);
        int strength = int.Parse(Line[2]);
        dataArray[i] = new StrengthLatLng(lat, lng, strength);
    }
    return dataArray;
}
```

Рис. 3.17 – Метод GetLineArray

Далі йде метод GetLineArray головна задача якого полягає, в обробці зміною в якій знаходяться дані отримані з файлу, та подача цих даних в вигляді одномірного масиву, класу StrengthLatLng розглянемо, як відбувається цей процес. Спочатку наші дані отримані з файлу йдуть, одним рядком, спочатку ми створюємо масив рядків, в який розбиваємо вхідний рядок по ідентифікатору '\n', який свідчить про перенос рядка, саме так в нас розділені дані про різні точки. Далі створюється одновимірний масив елементів класу StrengthLatLng, з кількістю елементів рівною кількості елементів в масиві рядків. Потім розпочинається цикл, в якому йде проходження по кожному рядку, в масиві рядків, він розбивається, ще на додаткові рядки по знаками ';', саме таким знаком

відділяються елементи однієї точки. Кожен з цих елементів присвоюється в відповідний їй тип даних з допомогою функції Parse, і в кінці ітерації, створюється нова точка, з переданими їй даними, отриманих з файла. В кінці виконання даного методу повертаємо значення масиву точок, як результат роботи методу.

Наступний метод GetBasicZoom, код якого зображено на Рис. 3.18.

```
public static int GetBasicZoom(StrengthLatLng[,] MatrixData, MercatorProjection projection)
{
    int zoom = 0;
    for (int i = 2; i <= 18; i++)
    {
        GPoint FirstPoint = projection.FromLatLngToPixel(MatrixData[MatrixData.GetLength(0) - 1, 0].pointLatLng, i);
        GPoint SecondPoint = projection.FromLatLngToPixel(MatrixData[0, MatrixData.GetLength(1) - 1].pointLatLng, i);
        int sWidth = (int)(SecondPoint.X - FirstPoint.X);
        int sHeight = (int)(SecondPoint.Y - FirstPoint.Y);
        if (sWidth <= MatrixData.GetLength(0) && sHeight <= MatrixData.GetLength(0))
        {
            zoom = i;
        }
    }
    return zoom;
}
```

Рис. 3.18 – Метод GetBasicZoom

Задача метода GetBasicZoom полягає в визначенні максимального зума, при якому в кожному пікселі буде один, або більше вимір, це потрібно тому що логіка побудови зображення в випадку коли на піксель припадає один вимір, або більше відрізняється від логіки побудови, коли на піксель не припадає ні одного виміру. Логіка роботи методу полягає в проходженні по кожному зуму, отримані кількості пікселей по довжині та висоті, та зрівняння цих даних з кількістю вимірів по довжині та ширині.

Розглянемо код методу. Спочатку створимо змінну, яку потім і будемо повертати в якості базового зуму. Після цього починається цикл, який буде проходити по всім доступним нам зумам, всередині циклу, створюється змінна зі значенням проекції верхнього лівого кута області побудови зображення, а після цього створюється змінна зі значенням проекції пікселя нижнього правого кута області побудови зображення. Далі знаючи вже координати верхнього лівого пікселя, і нижнього правого можна вирахувати, довжину і висоту зображення для зуму, потім ця довжина порівнюється з кількістю унікальних широт, та довготи відповідно, якщо ширина, і висота зображення менше, то змінній присвоюється

значення зуму, який тепер перевіряється, якщо ж більше то це значення не обновлюється більше, і потім повертається, як результат роботи методу.

Наступний метод GetPixelArray, код якого зображено на

```
public static StrengthPoint[,] GetPixelArray(StrengthLatLng[,] MatrixData, int zoom, MercatorProjection projection)
{
    GPoint StartPoint = projection.FromLatLngToPixel(MatrixData[MatrixData.GetLength(0) - 1, 0].pointLatLng, zoom);
    GPoint EndPoint = projection.FromLatLngToPixel(MatrixData[0, MatrixData.GetLength(1) - 1].pointLatLng, zoom);
    int width = (int)(EndPoint.X - StartPoint.X);
    int height = (int)(EndPoint.Y - StartPoint.Y);
    StrengthPoint[,] strengthPoints = new StrengthPoint[width + 1, height + 1];
    for (int i = 0; i < MatrixData.GetLength(0); i++)
    {
        for (int j = 0; j < MatrixData.GetLength(1); j++)
        {
            GPoint projectionPoint = projection.FromLatLngToPixel(MatrixData[i, j].pointLatLng, zoom);
            int x = (int)(projectionPoint.X - StartPoint.X);
            int y = (int)(projectionPoint.Y - StartPoint.Y);
            if (strengthPoints[x, y] == null)
            {
                strengthPoints[x, y] = new StrengthPoint();
                strengthPoints[x, y].Add(MatrixData[i, j].strength);
            }
            else
            {
                strengthPoints[x, y].Add(MatrixData[i, j].strength);
            }
        }
    }
    for (int i = 0; i < strengthPoints.GetLength(0); i++)
    {
        for (int j = 0; j < strengthPoints.GetLength(1); j++)
        {
            strengthPoints[i, j].Transform();
        }
    }
    return strengthPoints;
}
```

Рис. 3.19 – Метод GetPixelArray

Задача метода GetPixelArray полягає в перетворенні двовірного масиву точок, в форматі, широти, довготи, і значення напруженості точки, в двовірний масив точок, в форматі локальних координат пікселя по довжині та висоті на самому зображенні градієнта розподілу. Дані які приймаються при виклику метода, сам двовірний масив точок, в форматі, широти, довготи, і значення напруженості в точці, зум для якого це робиться, і об'єкт для створення проєкцій.

Розглянемо сам код, спочатку створюються дві змінні класу GPoint, які будуть містити в собі проєкцію крайньої лівої точки, та правої нижньої відповідно. Потім знаходиться з двох точок, довжина і висота зображення в пікселях. Далі створюється двовірний масив з розмірністю довжини і висоти, в пікселях. Потім за допомогою подвійного циклу обходимо двовірний масив точок, в форматі, широти, довготи, і значення напруженості в точці, по кожній

точці. Проектуємо точку з широти і довготи в значення глобального пікселя карти. Далі визначаємо значення локального пікселя, робимо це за допомогою, віднімання глобальних координат точки на карті, яка оброблюється, від значень глобальних координат точки лівого верхнього кута, і таким чином отримуємо локальні координати. Після цього перевіряється чи є масиві точок в форматі локальних координат, з індексами локальних координат даної точки об'єкт, якщо ні то створюємо і додаємо туди значення напруженості, якщо там то просто додаємо ще одне значення напруженості. Так як на одну точку може припадати більше ніж один вимір, то ми просто додаємо в комірку з пікселем всі значення напруженості, що були в них, а при наступному обході двовимірному масиву, вираховуємо середнє значення. В кінці повертаємо двовимірний масив, з значеннями напруженості в локальних точках, як результат роботи методу.

Наступний метод `getPictureFromPixelArray`, код якого зображено на Рис. 3.20

```
public static Bitmap getPictureFromPixelArray(StrengthPoint[,] pixelArray) {
    Bitmap bitmap = new Bitmap(pixelArray.GetLength(0), pixelArray.GetLength(1));

    Bitmap colorScheme = (Bitmap)Bitmap.FromFile(@"fire.png", true);
    for (int i = 0; i < pixelArray.GetLength(0); i++)
    {
        for (int j = 0; j < pixelArray.GetLength(1); j++)
        {
            Color pixel = colorScheme.GetPixel(0, pixelArray[i, j].strength);
            pixel = Color.FromArgb(pixelArray[i, j].strength, pixel);
            bitmap.SetPixel(i, j, pixel);
        }
    }
    return bitmap;
}
```

Рис. 3.20 – Метод `getPicturePixelArray`

Метою даного методу є побудова зображення, з двомірному масиву точок з локальними координатами, та значеннями напруженості в даній точці.

Розглянемо код. Спочатку створюється пусте зображення з довжиною розмірності двомірному масиву по довжині, та висотою розмірності двомірному масиву по висоті відповідно. Потім створюється палітра кольорів з файлу зображення, градієнта, який ми вибрали. Після цього за допомогою подвійного циклу обходимо кожен піксель. При обході певного пікселя, ми беремо



відповідне значення напруженості, та відповідно до нього беремо колір з палітри кольорів, після цього потрібно присвоїти ще значення альфа- каналу, або так званій прозорості, присвоїмо його теж відповідно до значення напруженості. Це означає, що зі зміною напруженості буде змінюватись не тільки колір, а й його прозорість. І забарвлюємо відповідний піксель цим кольором. Після цього повертаємо отримане зображення як результат роботи методу.

Наступний метод `getUpperPicture`, код якого зображено на Рис. 3.21.

```
public static Bitmap getUpperPicture(Bitmap picture, GPoint startPoint, GPoint endPoint) {
    int width = (int)(endPoint.X - startPoint.X);
    int height = (int)(endPoint.Y - startPoint.Y);
    Bitmap bitmap = new Bitmap(width, height);
    Graphics g = Graphics.FromImage(bitmap);
    g.InterpolationMode = System.Drawing.Drawing2D.InterpolationMode.HighQualityBicubic;
    g.DrawImage(picture, new Rectangle(0,0, width, height), 0, 0, picture.Width, picture.Height, GraphicsUnit.Pixel);
    return bitmap;
}
```

Рис. 3.21 – Метод `getUpperPicture`.

Задачею метода `getUpperPicture` є побудова зображення для таких значень приближень, на яких на один піксель може не припадати ні одного виміру.

Так як в нас на піксель може не припадати ні одного виміру, то ми мали б точки посеред зображення, які були б ніяк не забарвлені, при побудові тими самими методами, що і для зображень на яких на піксель припадає мінімум один вимір. Це було б не правильно, тому для рішення цієї проблеми, в даному методі застосовується інтерполяція зображення.

Розглянемо сам код. В метод `getUpperPicture` передається зображення з попереднього приближення, а також верхня ліва точка з глобальними координатами пікселя на карті, та права нижня точка, з глобальними координатами пікселя на карті. В самому методі спочатку вираховується довжина, та висота зображення для даного приближення. Далі створюється зображення з цією довжиною та висотою. Після цього створюється елемент, який відповідає за роботу з рисунням, і прив'язується до нашого зображення. Цьому елементу назначається властивість інтерполяції при рисуванні, після чого даємо команду на рисування зображення попереднього приближення, на більшому зображенні даного приближення, з використанням інтерполяція. В такому

випадку зображення попереднього приближення буде розтягнуто до розмірів теперішнього, а пусті пікселі забарвляться в колір який, буде обраховано вибраним нами методом інтерполяції. Готове зображення повертаємо, як результат роботи функції.

Тепер розглянемо додатково створенні мною класи.

```
using GMap.NET;

namespace StrengthElectrostaticFieldMap
{
    13 references
    public class StrengthLatLng
    {
        public PointLatLng pointLatLng;
        public int strength;
        0 references
        public StrengthLatLng(PointLatLng p, int s) {
            pointLatLng = p;
            strength = s;
        }
        1 reference
        public StrengthLatLng(double lat, double lng, int s) {
            pointLatLng = new PointLatLng(lat, lng);
            strength = s;
        }
    }
}
```

Рис. 3.22 –Клас StrengthLatLng

Клас StrengthLatLng був створений для зберігання точки в форматі координат широти та довготи на карті, а також напруженості в цій точці.

В даному класі містяться два поля перше це зберігання точки в форматі широти та довготи, що реалізується класом PointLatLng який міститься в бібліотеці GMap.NET, саме для цих цілей. І змінна типу цілого числа для зберігання значення напруженості.

В класі реалізовано два конструктори, в першому приймається два параметри це точка, яка вже являється об'єктом класу PointLatLng, та напруженості і дані напряму присвоюються полям класу. А в другому конструкторі приймається 3 значення, це значення широти окремо, значення довготи окремо, і напруженості. Тоді при присвоєні широти, та довготи викликається створення об'єкта відповідного класу.

```

using System.Collections.Generic;
namespace StrengthElectrostaticFieldMap
{
    8 references
    public class StrengthPoint
    {
        List<int> strengths = new List<int>();
        public int strength;
        2 references
        public void Add(int a)
        {
            strengths.Add(a);
        }

        1 reference
        public void Transform()
        {
            for (int i = 0; i < strengths.Count; i++)
            {
                strength += strengths[i];
            }
            strength /= strengths.Count;
        }
    }
}

```

Рис. 3.23 – Клас StrengthPoint

Клас StrengthPoint був створений для збереження напруженості. Використовується в програмі при створенні двомірного масиву локальних координат, де координатами виступають значення індексів, а сам тип зберігає тільки значення напруженості. Але так як буває, що на один піксель може припадати більше одного виміру, то потрібно було реалізувати можливість збереження масиву, при якому ми не знаємо скільки точно елементів, в нас буде в масиві. Це дозволяє зробити масив типу List. Він створюється пустий, а потім в нього додаються потрібні нам значення. Додавання значень в цей масив реалізовано за допомогою метода Add. А потім коли нам потрібно дізнатись середнє значення напруженості використовуємо метод Transform, який вираховує середнє арифметичне зі всіх значень напруженості, що потрапили в дану точку.

```

using System.Drawing;
using GMap.NET;

namespace StrengthElectrostaticFieldMap
{
    8 references
    public class ZoomCache
    {
        public GPoint startPoint;
        public Bitmap picture;
        public StrengthPoint[,] strengthPoints;

        1 reference
        public ZoomCache(GPoint startPoint, Bitmap picture, StrengthPoint[,] strengthPoints)
        {
            this.startPoint = startPoint;
            this.picture = picture;
            this.strengthPoints = strengthPoints;
        }

        1 reference
        public ZoomCache(GPoint startPoint, Bitmap picture)
        {
            this.startPoint = startPoint;
            this.picture = picture;
        }
    }
}

```

Рис. 3.24 – Клас ZoomCache

Клас ZoomCache був створений для зберігання зображення, точки прив'язки, та в деяких випадках двомірного масиву, зі значеннями напруженості та локальними координатами. В даному класі реалізовано два конструктора, перший приймає всі три значення, а другий тільки точку прив'язки, і зображення, це використовується в випадках з приближенням більше базового, так як в такому випадку ми не можемо обрахувати, двомірний масив.

### 3.3. Структурний аналіз програмного забезпечення

В ході розробки програмного забезпечення були використані деякі, технічні рішення, для прискорення роботи, та оптимізації ПЗ, для того, щоб користувачу було більш зручно ним користуватись.

Перше з таких рішень, це використання так званого кеша зображень. В даному випадку я створював свій клас додаткового шару, наслідуючись від базового класу додаткового шару, та заново визначав, його метод onRender, який викликається кожен раз, коли щось відбувається з графічною областю, тобто коли ми перетаскуємо карту, або змінюємо приближення, то відразу викликається цей метод, для нової побудови зображення об'єктів, відносно того, яка частина карти тепер відображається, для користувача. То наше зображення градієнта напруженості теж перебудовувалось би кожен раз, при зміні

положення карти. Що є дуже не зручним, так як зображення можуть бути великим і це буде визивати там звані ‘лаги’ чи ‘підтороможування’, що знижує її зручність використання. Тому було прийнято рішення при завантаженні даних з файлу відбувається їх опрацювання, і збереження масиву спеціального типу даних ZoomCache, готове зображення градієнта розподілу для кожного приближення і верхню ліву точки, тому що саме за неї ми будемо прив’язуватись до реальної карти. І в такому випадку при виклику функції onRender в нас просто буде підтягуватись з масиву, індексом якого буде виступати приближення, зображення, і точка прив’язки, після чого, викликається функція drawImage, яка малює дане зображення, з точки прив’язки.

Наступним кроком покращення програми стало додавання в нього асинхронного програмування. Так як картинка для всіх приближень буде готуватись заздалегідь то дана операція буде займати достатньо часу, тим більш якщо буде виміряно багато точок. В звичайному стилі програмування при виконанні, цієї складної задачі програма б просто ‘зависла’ на цей часі користувач нічого б не міг робити.

В нашому ж випадку застосовується асинхронний метод – в цьому випадку, всі задачі виконуються не по порядку, і не очікують закінчення одної, для початку іншої. В такій ситуації в нас може бути одна задача, яка запущена в асинхронному режимі, і сам комп’ютер буде по черзі розділяти ресурси комп’ютера на цю задачу, і на синхронний потік, якщо це потрібно буде.

Для оголошення, що задача буде виконана в асинхронному режимі буде використовуватись спеціальна команда async, await.

```
private async void button1_Click(object sender, EventArgs e)
{
    if (openFileDialog1.ShowDialog() == DialogResult.Cancel)
        return;
    string filename = openFileDialog1.FileName;
    using (var sr = new StreamReader(filename)) {
        Data = sr.ReadToEnd();
    }
    label1.Text = "Очікуйте йде обробка даних";
    await Task.Run(() => sMapOverlay = new StrengthMapOverlay("Karpaty", gMap, Data));
    button2.Enabled = true;
    label1.Text = "Обробка закінчена.\nНатисніть відобразити результат.";
}
```

Рис. 3.25 Приклад Async

Параметр `async` в оголошенні методу, означає що деякі з даних функцій методу можуть бути виконані асинхронно. Параметр `await` вказує яка саме функція буде виконуватись в асинхронному режимі.

Та третьою особливістю є використання інтерполяції зображень, для обробки зображень градієнта розподілу напруженості.

В C# існує влаштована функція інтерполяції зображень, і він передбачає такі методи інтерполяції зображень:

- `Bicubic` - Визначає бікубічну інтерполяцію. Попередня фільтрація не проводиться. Цей режим не підходить для зменшення розміру зображення нижче 25 відсотків від його початкового розміру.
- `Bilinear` - Визначає білінійну інтерполяцію. Попередня фільтрація не проводиться. Цей режим не підходить для зменшення розміру зображення нижче 50 відсотків від його початкового розміру.
- `Default` - Визначає режим за замовчуванням.
- `High` - Визначає високу якість інтерполяції.
- `HighQualityBicubic` - Визначає високоякісну бікубічну інтерполяцію. Для забезпечення якісної усадки проводиться попередня фільтрація. Цей режим створює трансформовані зображення найвищої якості.
- `HighQualityBilinear` - Визначає високоякісну білінійну інтерполяцію. Для забезпечення якісної усадки проводиться попередня фільтрація.
- `Low` - Визначає інтерполяцію низької якості.
- `NearestNeighbor` - Визначає інтерполяцію найближчого сусіда.

Також в майбутньому можна буде додати серіалізацію, та десеріалізацію, в файл, для можливості збереження вже оброблених даних. А потім їх завантаження, без необхідності повторної обробки, що також зекономить нам час.

### 3.4. Використання програми

Після старту програми ми побачимо наступне вікно:

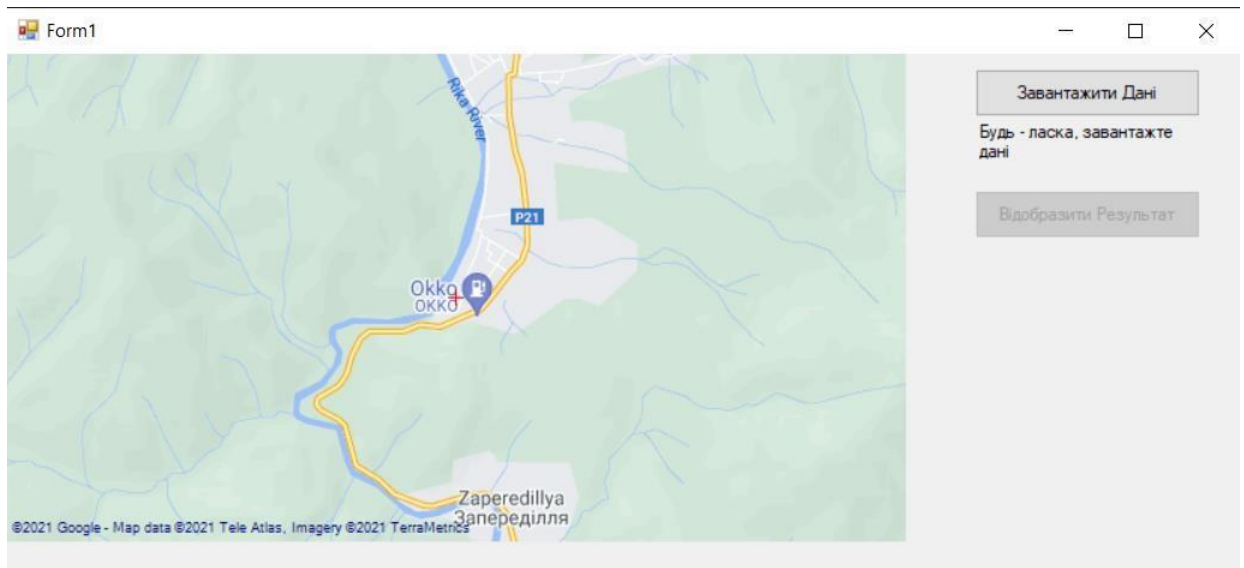


Рис. 3.26 – Головне вікно програми

На якому потрібно натиснути кнопку ‘Завантажити дані’, про що нам і говориться в вікні повідомлень програми. Після натискання на кнопку відкриється вікно вибору файлу:

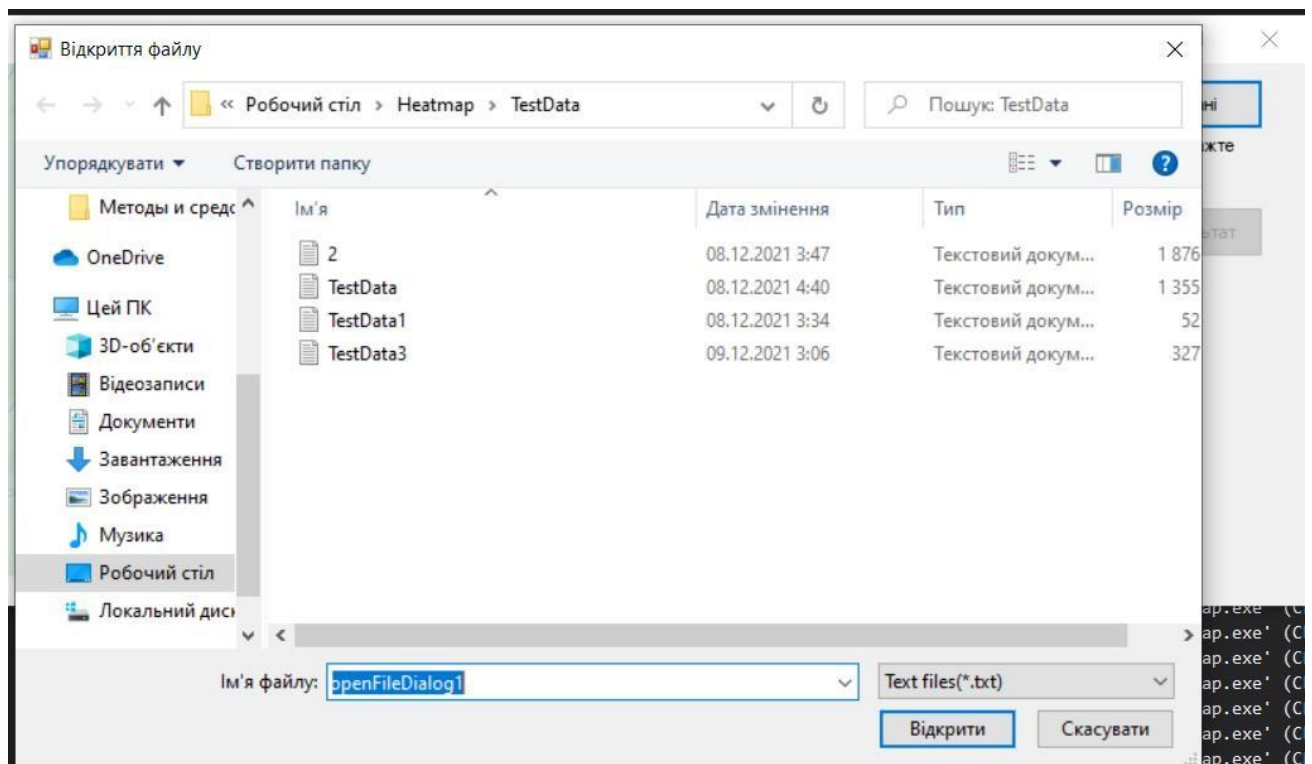


Рис. 3.27 – Вікно вибору файлу

Після того як ми відкриємо файл, програма почне обробляти завантажені нами дані, про що виведе відповідне повідомлення, в вікно повідомлень програми:

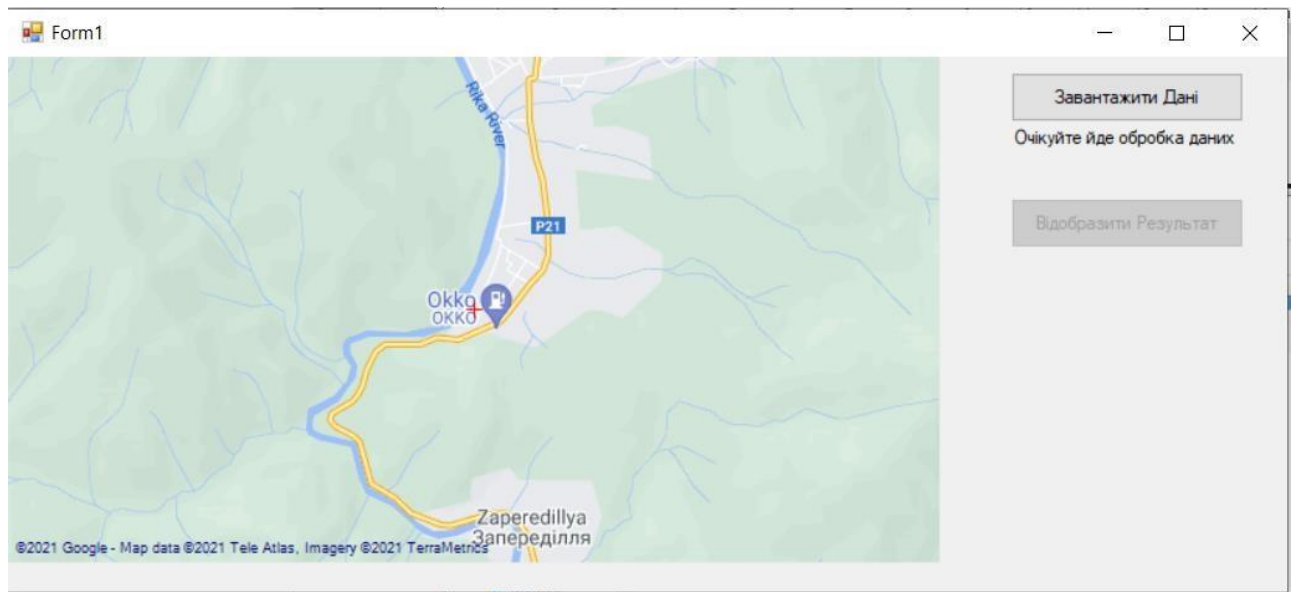


Рис. 3.28 – Вікно програми при обробці даних

Після обробки даних в вікно повідомлень виводиться повідомлення, про завершення обробки, та стає активна друга кнопка 'Відобразити результат':

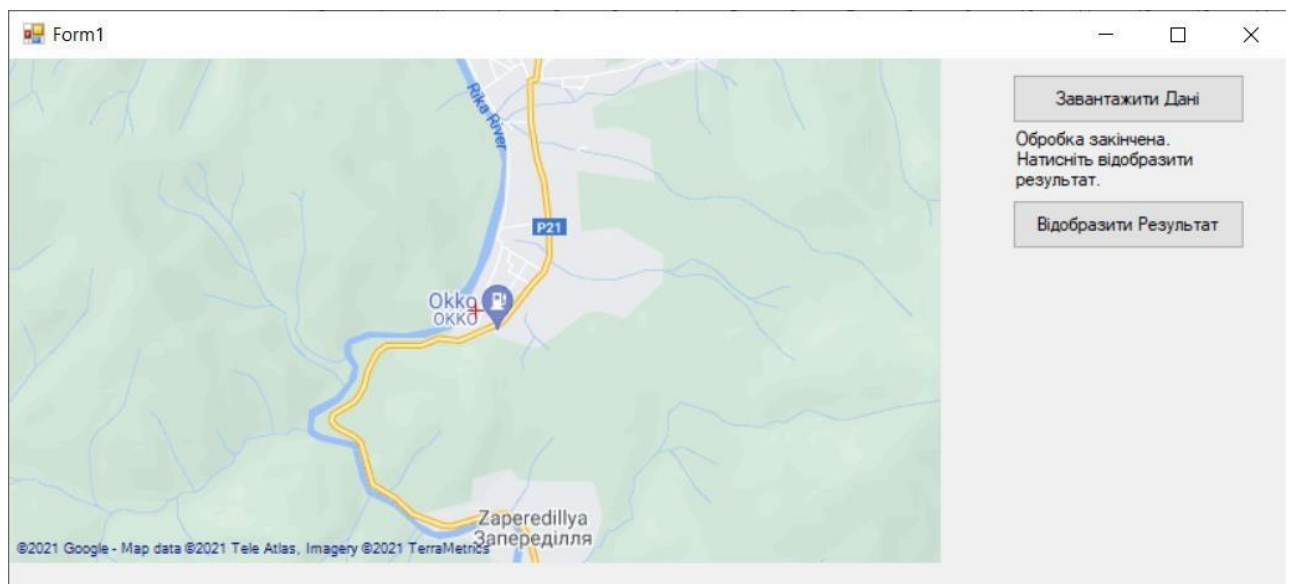


Рис. 3.29 – Вікно програми після обробки даних.

Натискаємо на кнопку 'Відобразити результат', і бачимо нанесений градієнт розподілу.





Рис. 3.30 – Виведені результати

### 3.5. Висновки до розділу

В даному розділі була проведена розробка ПЗ, для автоматизованої системи вимірювання напруженості електричного поля атмосфери на місцевості.

В якості мови програмування була обрана мова програмування C#. Так як за її використання можна створювати програми для платформи Windows, які не будуть потребувати скачування додаткового ПЗ, такого як інтерпретатори. А також в майбутньому на базі даної платформи є можливим розробити дане програмне забезпечення для платформи Android і IOS.

Був проведений аналіз можливостей мови і готових бібліотек, які надають змогу розробити програмне забезпечення даного типу. Також була проведена сама розробка цього ПЗ, та її тестування.

## 4. РОЗРОБКА СТАРТАП-ПРОЄКТУ «АВТОМАТИЗОВАНИЙ ПОРТАТИВНИЙ ПРИЛАД ДЛЯ ВИМІРЮВАННЯ НАПРУЖЕНОСТІ ЕЛЕКТРОСТАТИЧНОГО ПОЛЯ НА МІСЦЕВОСТІ»

### 4.1. Опис технологічний аудит ідеї проєкту

Стартап базується на ідеї портативного приладу малих габаритів для вимірювання напруженості електростатичного поля атмосфери на місцевості. Даний прилад може бути використаний в георозвідці, при первинному пошуку корисних копалин, а також створення первинних контурів залягання корисних копалин. Також може знайти застосування для наукових цілей, так як напруженість, та всі фактори впливання на неї не вивчені. Головною особливістю та відмінністю являється використання модулів GPS, та SD-card, що дає змогу робити вимірювання в автоматичному режимі, а при використанні квадрокоптеру, надає змогу виключити людину з процесі вимірювання. Також особливістю даної системи є використання датчика динамічного типу, який є досить швидким, невеликих розмірів, та не дорогим. Ще однією метою є створення власного ПЗ для простого та швидкого відображення даних вимірювання, та допомагає зробити продукт більш незалежним, що уточнено наведено в Таблиця 4.1.

В даному розділі буде проведено аналіз стартап-проєкту головною метою якого є визначення чи зможе наш продукт вийти на ринок та чи зможе він скласти конкуренцію які вже є в цій ніші та займають пріоритетні місця в ній.

Таблиця 4.1. Опис ідеї стартап - проєкту

<i>Зміст ідеї</i>	<i>Напрямки застосування</i>	<i>Вигода для користувача</i>
Основна ідея, це розробка автоматизованого приладу для вимірювання напруженості електричного поля атмосфери на певній місцевості	Георозвідка на предмет корисних копалин	- Мала вартість - Зручний формат відображення результатів - Простий в використанні
	Для наукових досліджень	

Загалом, в стартап проєкті пропонується повністю новий прилад аналогів якому, поки не існує, суть якого полягає в вимірюванні такого параметру як

напруженість електростатичного поля, на якійсь певній місцевості з зберіганням значень вимірювання в кожній точці місцевості, і координат цієї точки в файл що знаходить на SD карті, після чого можна загрузити даний файл в спеціально розроблену нами програму, та отримати градієнт розподілу на реальній карті

Далі проведемо аналіз потенційних техніко-економічних переваг порівняно з пропозиціями найближчих конкурентів:

– визначимо перелік техніко-економічних властивостей та характеристик ідеї;

– визначаємо коло найближчих конкурентів (проектів-конкурентів) або товарів-аналогів, або товарів-замінників, що вже існують на ринку, та проводимо збір інформації щодо значень техніко-економічних показників для ідеї власного проекту та проектів-конкурентів відповідно до визначеного вище переліку

– проводимо порівняльний аналіз показників: для власної ідеї визначено показники, що мають а) гірші значення (W, слабкі); б) аналогічні (N, нейтральні) значення; в) кращі значення (S, сильні) (Таблиця 4.2).

Конкуренти:

Конкурент1 - Tenmars TM-190

Конкурент2 - Warmbier 7100.EFM51

Таблиця 4.2.Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№	Техніко-економічні Характеристики ідеї	Товари/концепції конкурентів			W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проєкт	Конкурент 1 Tenmars TM-190	Конкурент 2 Warmbier 7100.EFM51			
1	Вартість, грн	3000	5300	31360			+
2	Вимірювання малих значень	Може вимірювати малень	Немає такого функціонала, тільки	Може вимірювати і маленькі значення		+	

		кі значен ня	промисло ві значення				
3	Можливість запису значень	Так на SD-card	Ні	Ні			+
4	Можливість передачі даних на ПК	Так, через SD-card	Немає	Немає			+
5	Автоматизоване вимірювання	Є	Немає	Немає			+

З Таблиця 4.2 можна зробити висновки, що дана система не тільки має переваги в базових можливостях приладів конкурентів, а має й велику кількість переваг виражених в можливостях збереження даних вимірювання, й можливість автоматизованого вимірювання.

В межах даного підрозділу проводимо аудит технології, за допомогою якої можна реалізувати ідею створення проекту.

Визначення технологічної здійсненності ідеї проекту передбачає аналіз складових які вказані в Таблиця 4.3.

Таблиця 4.3. Технологічна здійсненість ідеї проекту

<i>№</i>	<i>Ідея проекту</i>	<i>Технології реалізації</i>	<i>Наявність технологій</i>	<i>Доступність технологій</i>
1	Система автоматизованого вимірювання напруженості електричного поля на місцевості	Виготовлення вимірювача під замовлення	Наявна	Недоступна
		Самостійне виготовлення вимірювача використовуючи ЧПУ, та наявні електричні компоненти	Наявна	Доступна
2	Створення програмного забезпечення для обробки результатів вимірювання	Мова програмування C#	Наявна	Доступна

Обрана технологія реалізації ідеї проекту: було обрано самостійне виготовлення вимірювача використовуючи ЧПУ, та наявні електричні компоненти, для вимірювача, та мова програмування С# для створення ПЗ

Результати даного аналізу проекту показують, що за наявних технологій ми маємо можливість самостійно виготовити вимірювач, а також самостійно створити ПЗ, що зробить ціну системи значно меншою.

#### **4.2. Аналіз ринкових можливостей запуску стартап-проекту**

В даному підрозділі визначаємо ринкові можливості, які можна використати під час ринкового впровадження проекту, та ринкові загрози, які можуть перешкодити його реалізації. Визначення ринкових можливостей дозволяє спланувати напрям розвитку, потреб потенційних клієнтів та пропозицій проектів-конкурентів. Спочатку проведемо аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку.

Особливих конкурентів на ринку поки не існує, так як більшість існуючих приборів, або застарівші, або використовуються, як звичайні вимірювачі, та не мають ніякої автоматизації

Таблиця 4.4. Попередня характеристика потенційного ринку стартап-проекту

<i>№</i>	<i>Показники стану ринку (найменування)</i>	<i>Характеристика</i>
1	Кількість головних гравців, од	2
2	Загальний обсяг продаж, грн	Невідомий
3	Динаміка ринку	Стабільна
4	Наявність обмежень для входу	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	100

Аналіз Таблиця 4.4 показує, що ринок має стабільну динаміку і хороший попит на запропонований нами продукт, тому є доцільним вихід на ринок. Визначимо потенційних клієнтів, котрим будемо пропонувати нашу систему.

Таблиця 4.5. Характеристика потенційних клієнтів стартап-проекту

<i>№</i>	<i>Потреба, що формує ринок</i>	<i>Цільова аудиторія (цільові сегменти ринку)</i>	<i>Відмінності у поведінці різних потенційних цільових груп клієнтів</i>	<i>Вимоги споживачів до товару</i>
1	Потреба в недорогому автоматизованому вимірювачі	Наукові організації. Організації георозвідки по пошуку корисних копалин	Орієнтація на малогабаритні автоматизовані вимірювачі, з можливістю виводів результатів вимірювання в зручному форматі для подальшого аналізу	- малі розміри -автоматизоване вимірювання -подання результатів в зручному форматі

В даній таблиці ми описали фактори, котрі формують поведінку користувачів відносно нашої системи та їх основні вимоги до продукції.

При застосуванні даної технології існують певні загрози, тож буде доцільно провести аналіз факторів ризику та можливостей при виведенні товару на ринок.

Таблиця 4.6. Фактори загроз

<i>№</i>	<i>Фактор</i>	<i>Зміст загрози</i>	<i>Можлива реакція компанії</i>
1	Клієнтська база	Ринок даних проборів ще не сформований, тому набрати початкову клієнтську базу буде дуже складно	Проведення рекламної компанії серед цільової аудиторії
2	Відсутність навичок використання інноваційних систем даного типу	Так як система на даний момент немає аналогів, і до цього ніхто нічого схожого не використовував, то можуть бути складності з першими використаннями	Створення детальних інструкцій для користувачів
3	Термін роботи	Так як датчики даного типу майже ніде не використовувались, то немає ніякої інформації про його	Постійний технічний моніторинг

		зносостійкість, та терміни роботи	
4	Відсутність початкового попиту	Не бажання переходити на нові технології	Приведення аргументованих доказів, що зміна технології приведе до збільшення попиту на продукт.
5	Конкуренція	Можливий вихід на ринок іноземних компаній конкурентів	Постійне вдосконалення системи

В Таблиця 4.6. ми визначили можливі фактори загроз, такі як: клієнтська база, відсутність навичок використання інноваційних систем даного типу, термін роботи, відсутність початкового попиту, конкуренція. Вони в майбутньому можуть перешкоджати ринковому впровадженню нашого проекту, а також можливу реакцію компанії на загрози та шляхи їх усунення.

Але на одному рівні з загрозами є також і можливості.

Таблиця 4.7. Фактори можливостей

<i>№</i>	<i>Фактор</i>	<i>Зміст можливостей</i>	<i>Можлива реакція компанії</i>
1	Відсутність даного типу систем	На даний момент не існує схожих систем, з можливостями автоматизованого виміру напруженості електростатичного поля атмосфери на місцевості	Акцентування уваги клієнтів на інноваційних факторах
2	Наявність власного ПЗ	В процесі розробки стартапу була розроблена власна програма для обробки результатів вимірювання	Акцентування уваги на самостійності стартапу, та не потрібності покупки додаткового ПО
3	Подача результатів вимірювання в форматі легкому для подальшого аналізу	В ПЗ передбачено вивід інформації накладанням шару з градієнтом розподілу напруженості на реальну карту місцевості, а також можливість окремого збереження цього	Демонстрування даного фактору в рекламній компанії

		зображення, що робить його дуже інформативним для аналізу.	
4	Участь в виставках	Так як продукт новий та інноваційний маємо можливість участі в виставках з найсучаснішим обладнанням	Розповсюдження рекламної продукції, даної системи
5	Створення нової ніші	Так як прилад новий, то можна створити нову нішу, і відразу зайняти в ній ключову позицію	Активне закріплення в створеній ніші, для створення сильної конкурентно-спроможності перед майбутніми конкурентами

В Таблиця 4.7. були визначені фактори можливостей такі як: відсутність даного типу систем, наявність власного ПЗ, подача результатів вимірювання в форматі легкому для подальшого аналізу, участь в виставках, створення нової ніші. Вони посприяють ринковому впровадженню даного проекту.

Також доцільно буде провести аналіз ринку для подальшого формування стратегії.

Таблиця 4.8. Ступеневий аналіз конкуренції на ринку

<i>Особливості конкурентного середовища</i>	<i>В чому проявляється дана характеристика</i>	<i>Вплив на діяльність підприємства(можливі дії компанії, щоб бути конкурентноспроможною)</i>
1. Тип конкуренції - Олігополія	Існує декілька схожих приладів.	Дана система є автоматизована, тому є сильним конкурентом, для приладів, які вже є на ринку.
2. Міжнародний	В середині країни геодезичні розслідування на предмет корисних копалин, є не сильно поширеними.	Вихід на міжнародний ринок.
3. Міжгалузєва	Система може використовуватись в галузях: пошуку корисних копалин і в галузі наукових досліджень	Вдосконалення продукту, яке не має прив'язки до галузі



4. Конкуренція за видами товарів, товарно-видова	Конкуренція між товарами одного виду	Система є автоматизована на відміну від конкурентів, що робить її більш привабливою для користувача
5. За характером конкурентних переваг: цінова та нецінова	Цінова полягає в тому, що прилад коштує менше ніж в конкурентів, а не цінова в тому що прилад набагато якісніший, та автоматизований	Цінова- ключові деталі системи ми будемо виготовляти самі, так як це власна розробка. Нецінова – автоматизована система, з власним ПЗ
6. За інтенсивністю: не марочна	Бренди відсутні	

Судячи з даної таблиці можна зробити висновок, що ситуація на ринку характеризується слабкою конкуренцією, тому відсутні фактори, котрі сильно обмежують входження в ринок.

Більш детальний аналіз умов конкуренції наведено в Таблиця 4.9.

Таблиця 4.9. Аналіз в конкуренції в галузі за М. Портером

	<i>Прямі конкуренти</i>	<i>Потенційні конкуренти</i>	<i>Постачальники</i>	<i>Клієнти</i>	<i>Товари-замінники</i>
<i>Складові аналізу</i>	Немає прямих конкурентів	Є два прилади схожих, які в майбутньому можуть стати прямими конкурентами	Постачальниками є спеціальні компанії, які займаються розповсюдженнями приладів даного типу	Основними клієнтами є наукові підприємства, та підприємства які займаються пошуком корисних копалин	Товарів – замінників немає
<i>Висновки :</i>	Конкуренції немає	Постійне вдосконалення приладу, щоб потенційні конкуренти не могли стати прямими конкурентами	Постачальники не диктують умови роботи на ринку	Для клієнтів вагомим питанням є ціна, та функціональність і зручність.	Конкуренції немає

Провівши аналіз таблиці можна зробити висновок, що з огляду на конкурентну ситуацію на ринку можливість роботи на ринку присутня. Також були визначені характеристики, які повинен мати проект і які дії повинна

проводити компанія, щоб бути конкурентоспроможною на ринку.

Після аналізу визначається та обґрунтовується перелік факторів що впливають на конкурентоспроможність. Допоки проект не впроваджено в життя, цей аналіз важко провести точно, можна дати тільки попередню оцінку конкурентоспроможності.

Основаючись на аналізі конкуренції, а також враховуючи характеристики ідеї проекту, вимоги споживачів до товару, а також фактори маркетингового середовища, можна визначити перелік факторів конкурентоспроможності. Аналіз факторів приведений в наступній Таблиця 4.10..

Таблиця 4.10. Обґрунтування факторів конкурентоспроможності

<i>№ n/n</i>	<i>Фактор конкурентоспроможності</i>	<i>Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)</i>
1	Новизна	Товар не має прямих аналогів на ринку
2	Ціновий	Невисока вартість
3	Автоматизованість	За рахунок наявності GPS та SD-card
4	Точність	Прилад має хорошу точність

В даній таблиці проведено на основі Таблиця 4.9 визначили та обґрунтували фактори конкурентоспроможності даного проекту.

За попередньо визначеними факторами конкурентоспроможності проведемо аналіз слабких та сильних сторін проекту.

Таблиця 4.11. Порівняльний аналіз сильних та слабких сторін «АВТОМАТИЗОВАНИЙ ПОРТАТИВНИЙ ПРИЛАД ДЛЯ ВИМІРЮВАННЯ НАПРУЖЕНОСТІ ЕЛЕКТРОСТАТИЧНОГО ПОЛЯ НА МІСЦЕВОСТІ»

<i>№</i>	<i>Фактор конкурентоспроможності</i>	<i>Бали 1-20</i>	<i>Рейтинг у порівнянні з конкурентами</i>						
			-3	-2	-1	0	+1	+2	+3
1	Новизна							+	
2	Ціна							+	
3	Автоматизованість								+
4	Точність					+			

З Таблиця 4.9 та Таблиця 4.10. бачимо, що фактори конкурентоспроможності суттєві та мають великий позитивний внесок при

впровадженні нових елементів. Основною перевагою та головним досягненням є автоматизованість.

Таблиця 4.12. SWOT- аналіз стартап-проекту

<p>Сильні сторони:</p> <ol style="list-style-type: none"> <li>1. Вартість</li> <li>2. Вимірювання малих значень</li> <li>3. Можливість запису значень</li> <li>4. Можливість передачі даних на ПК</li> <li>5. Автоматизоване вимірювання</li> </ol>	<p>Слабкі сторони:</p> <ol style="list-style-type: none"> <li>1. Продукт ще не зарекомендував себе на ринку</li> <li>2. Мале поширення технології і як наслідок недовіра до неї</li> </ol>
<p>Можливості:</p> <ol style="list-style-type: none"> <li>1. Участь в виставках</li> <li>2. Створення нової ніші</li> <li>3. Заняття високої позиції в створеній ніші</li> <li>4. Окреме просування ПЗ для схожих продуктів</li> </ol>	<p>Загрози:</p> <ol style="list-style-type: none"> <li>1. Клієнтська база</li> <li>2. Відсутність навичок використання інноваційних систем даного типу у можливих користувачів</li> <li>3. Термін роботи</li> </ol>

В Таблиця 4.12. проводимо перелік сильних та слабких сторін, а також ринкових загроз та ринкових можливостей, який складаємо на основі факторів загроз і можливостей, який ми склали раніше. Ринкові загрози та можливості на відміну від факторів ще не є реалізованими на ринку та мають певну ймовірність здійснення.

На основі SWOT-аналізу розробляємо альтернативи ринкової поведінки для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок.

Визначені альтернативи аналізуються з точки зору строків та ймовірності отримання ресурсів.

Таблиця 4.13. Альтернативи ринкового впровадження стартап-проекту

№	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Стратегія нейтралізації ринкових загроз сильними сторонами стартапу	Висока	2 роки
2	Стратегія компенсації слабких сторін стартапу наявними ринковими можливостями	Висока	1.5 рік
3	Стратегія виходу з ринку	Низька	

### 4.3. Розроблення ринкової стратегії продукту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку. Для цього проведемо опис цільових груп потенційних користувачів.

Таблиця 4.14. Вибір цільових груп потенційних споживачів

<i>№</i>	<i>Опис профілю цільової групи потенційних клієнтів</i>	<i>Готовність споживачів сприйняти продукт</i>	<i>Орієнтовний попит в межах цільової групи (сегменту)</i>	<i>Інтенсивність конкуренції в сегменті</i>	<i>Простота входу у сегмент</i>
1	Наукові підприємства	Середня	80%	30%	Середня
2	Підприємства що займаються пошуком корисних копалин	Середня	90%	60%	Середня
Які цільові групи обрано: Наукові підприємства, та підприємства що займаються пошуком корисних копалин					

Виходячи з результатів аналізу потенційних груп користувачів ми обираємо такі цільові групи, як наукові підприємства, та підприємства що займаються пошуком корисних копалин. Саме для них ми будемо пропонувати нашу автоматизовану систему для контролю поверхневих дефектів в об'єктах, що мають складну геометричну форму.

Для роботи в даному сегменті ринку потрібно сформувати базову стратегію розвитку.

Таблиця 4.15. Визначення базової стратегії розвитку

<i>№</i>	<i>Обрана альтернатива розвитку проекту</i>	<i>Стратегія охоплення ринку</i>	<i>Ключові конкурентоспроможні позиції відповідно до обраної альтернативи</i>	<i>Базова стратегія розвитку*</i>
1	Вдосконалення	Диференційний маркетинг	Автоматизація, зручність, якість, ціна, габарити	Стратегія диференціації

За базову стратегію розвитку було взято стратегію диференціації, що передбачає надання товару важливих, з точки зору споживача, відмінних

властивостей, які роблять систему, яка відрізняється від тієї, що є в конкурентів

Наступним кроком є вибір стратегії конкурентної поведінки(Таблиця 4.16. ).

Таблиця 4.16. Визначення базової стратегії конкурентної поведінки

<i>№</i>	<i>Чи є проект «першопрохідцем» на ринку?</i>	<i>Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?</i>	<i>Чи буде компанія копіювати основні характеристики товару конкурента, і які?</i>	<i>Стратегія конкурентної поведінки*</i>
1	Так	Компанія буде шукати нових споживачів	Не буде копіювати основні характеристики	Стратегія заняття конкурентної ніші

За базову стратегію конкурентної поведінки була прийнята стратегія зайняття конкурентної ніші, коли компанія в якості цільового ринку вибирає один або декілька ринкових сегментів малого розміру. Головне завдання компанії при цьому – це постійна підтримка та розвиток своєї конкурентної переваги.

На основі вимог споживачів з обраного сегменту до постачальника і продукту, а також в залежності від стратегії розвитку та стратегії конкурентної поведінки розробляємо стратегію позиціонування яка визначається у формування ринкової позиції, за яким споживачі мають ідентифікувати проект

Таблиця 4.17. Визначення стратегії позиціонування

<i>№</i>	<i>Вимоги до товару цільової аудиторії</i>	<i>Базова стратегія розвитку</i>	<i>Ключові конкурентоспроможні позиції власного стартап-проекту</i>	<i>Вибір асоціацій, які мають сформувану комплексну позицію власного проекту (три ключових)</i>
1	Автоматизованість, та вдосконалення в залежності від вимог споживача	Стратегія диференціації	Автоматизація, зручність, якість, ціна, габарити	Зручність, автоматизованість, вивід результатів в зручному форматі

Результатом даного підрозділу є система рішень щодо ринкової поведінки компанії, вона визначає в якому напрямі буде працювати компанія на ринку.

#### 4.4. . Розроблення маркетингової програми стартап-проекту

Під час розроблення маркетингової програми першим кроком є формулювання маркетингової концепції товару, котру отримає користувач.

Для цього підсумуємо результати попереднього аналізу конкурентоспроможності товару.

Таблиця 4.18. Визначення ключових переваг концепції потенційного товару

<i>№</i>	<i>Потреба</i>	<i>Вигода, яку пропонує товар</i>	<i>Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)</i>
1	Ціна	Низька ціна, за рахунок власного виготовлення ключових компонентів	Збільшення кількості клієнтів
2	Автоматизація	Із-за присутності в системі GPS, не система автоматично запам'ятовує в якій координаті, яке було виміряне значення	Зручність для користувача
3	Інтеграція з ПК, через власне ПЗ	Зручне передавання виміряних даних за допомогою SD-card, та інформативна подача результатів	Автоматизована обробка результатів вимірювання, і подача їх в зручному форматі

В даній таблиці формуємо комплекс асоціацій за якими споживачі будуть ідентифікувати наш торгівельний проект.

Результатом даного підрозділу є система рішень щодо ринкової поведінки компанії, вона визначає в якому напрямі буде працювати компанія на ринку.

Розробимо трирівневу маркетингову модель товару: уточняється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (Таблиця 4.19.).

Таблиця 4.19. Опис трьох рівнів моделі товару

<i>Рівні товару</i>	<i>Сутність та складові</i>		
<i>I. Товар за задумом</i>	<i>Автоматизована система вимірювання напруженості електростатичного поля атмосфери на місцевості</i>		
<i>II. Товар у реальному виконанні</i>	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Довговічність	Нм	Тх
	2. Гарантійний термін	Нм	Е
	3. Досконалість виробничого виконання	Нм	Тл
	4. Вартість обслуговування	М	Вр
	5. Технологічність	М	Тх
	Якість: стандарти		
	Пакування: коробка		
	Марка: назва організації розробника		
	<i>III. Товар із підкріпленням</i>	До продажу: Демонстрація можливостей приладу	
Після продажу: після продажне обслуговування			
За рахунок чого потенційний товар буде захищено від копіювання: патент, свідоцтво на авторське право			

В Таблиця 4.19. ми створюємо трьохрівневу модель нашого товару що включає задум товару та його вигоди, основі характеристики готового товару, спосіб його пакування та захисту від копіювання та плагіату.

Визначимо цінові межі, якими необхідно керуватись при встановленні ціни на потенційний товар, яке передбачає аналіз ціни на товари-аналоги або товари субститути, а також аналіз рівня доходів цільової групи споживачів (

Таблиця 4.20.). Аналіз проводиться експертним методом.

Таблиця 4.20. Визначення меж встановлення ціни

<i>№</i>	<i>Рівень цін на товаризамінники</i>	<i>Рівень цін на товари аналоги</i>	<i>Рівень доходів цільової групи споживачів</i>	<i>Верхня та нижня межі встановлення ціни на товар/послугу</i>
1	20 тис	5 тис – 30 тис	100 тис- 200 тис	2 тис – 5 тис

Таблиця 4.20. проведений аналіз ринкової ціни на товари аналоги та товари заміники, а також середній рівень доходів користувачів. За одержаними даними встановлена верхня ти нижня границя на наш проект.

Визначимо оптимальну систему збуту, в межах якого приймається рішення.

Таблиця 4.21. Формування системи збуту

<i>№</i>	<i>Специфіка закупівельної поведінки цільових клієнтів</i>	<i>Функції збуту, які має виконувати постачальник товару</i>	<i>Глибина каналу збуту</i>	<i>Оптимальна система збуту</i>
1	Роздрібна продаж	Продаж клієнтам продукції	Висока	Проведення збуту силами фірм посередників, та власними силами

Було проведено розробку концепцію маркетингових комунікацій між користувачами та виробником. В нашому випадку, дана робота з виробником та реалізація продукту через прямий продаж.



Розробимо концепцію маркетингової комунікації, котра опирається на попередньо обрану основу для позиціонування, обрану специфіку поведінки користувачів.

Таблиця 4.22. Концепція маркетингових комунікацій

<i>№</i>	<i>Специфіка поведінки цільових клієнтів</i>	<i>Канали комунікацій якими користуються цільові клієнти</i>	<i>Ключові позиції, обрані для позиціонування</i>	<i>Завдання рекламного повідомлення</i>	<i>Концепція рекламного звернення</i>
1	Пошук в інтернеті через спеціальні ресурси, та нові патенти	Інтернет, прямий зв'язок з виробниками	Автоматизація, ціна, якість	Акцентування уваги клієнтів, на широкі можливості, автоматизацію процесу	Ціна, автоматизація

Спираючись на специфіку закупівельної поведінки цільових клієнтів було обрано власну систему збуту, коли виробник безпосередньо продає товар клієнту. Ціноутворення відбувається на основі аналізу товарів – аналогів і відбувається під час фінансово-економічного аналізу проекту з залученням експертів.

#### 4.5. Організація реалізації стартап-проекту

Представимо команду стартап-проекту та її розвиток

Таблиця 4.23. Команда стартап-проекту

<i>Розвиток команди</i>	<i>Учасники</i>	<i>Завдання учасників в</i>	<i>Освіта</i>	<i>Досвід роботи</i>	<i>Спеціалізовані знання</i>	<i>Витрати, тис. грн.</i>

Мінімальна команда	Белак Артур	Керівник; розробни; бухгалтер	Бакалавр	Інженер; розробни; електрик; програміст	Наявні певні знання в розробці інтелектуальних систем, створенні різних інженерних рішень, знання електроніки та електрики, знання теорії програмування, та мови програмування C#	10000
Разом:						10000

Дана таблиця представляє повну команду розробників системи, що являється мінімальною життєздатною командою, але дана команда без зволікань може повністю представляти інтереси стартап-проекту

Розробимо календарний графік реалізації стартап-проекту.

Таблиця 4.24. Календарний графік реалізації стартап-проекту

Зміст етапу	Період реалізації, номер місяця з початку виконання робіт													Вартість етапу, грн.	
	1	2	3	4	5	6	7	8	9	10	11	12	13		
Розробка системи	■	■													2000
Визначення кошторису			■												5000
Проектування системи				■	■	■									30000
Проектування конструкції						■									25000
Створення програмного забезпечення							■	■	■						23000

Створення прототипу																		10000
Тестування																		7500
Виявлення та усунення недоліків																		15000
Запуск реклами																		10000
Запуск виробництва																		50000
Всього, грн																	177500	

Таблиця 4.24. представляє собою календарний план-графік реалізації стартап проекту, в якій передбачені всі можливі витрати з початку розробки аж до самого виробництва, в гривнях.

Визначимо ціни на сировину, матеріали та комплектуючі виробу та визначимо постачальників

Таблиця 4.25. Сировина, матеріали та комплектуючі виробу

<i>№</i>	<i>Найменування видів сировини, матеріалів та комплектуючих виробів</i>	<i>Постачальники</i>	<i>Ціна за одиницю</i>	<i>Примітки (місцезнаходження постачальників, умови поставок тощо)</i>
1.	Вимірювач напруженості	Виготовляємо самі	500	Виготовляємо самостійно
2.	STM32	Інвестори можуть бути постачальниками	100	Постачальники знаходяться в Китаї; поставки виконуються за замовленням
4.	Корпусні деталі		300	
5.	GPS		250	
6.	SD-Card		150	

7.	Плати	100
Разом:		1400

В даній таблиці прописані матеріали, що являються необхідними для розробки системи описаної даним стартап-проектом. Враховуючи те що в даній таблиці вказана роздрібна ціна, то при масовому виробництві матеріали будуть постачатись по оптовим цінам і в результаті затрати будуть декілька меншими.

Таблиця 4.26. Потреба в промислово-виробничому персоналі

№	Посада/виконувані завдання	Чисельність	Витрати на персонал, тис. грн.
1.	Керівник	1	15000
2.	Бухгалтер	1	10000
3.	Інженер	1	10000
4.	Інженер-електронщик	1	10000
6.	Програміст	1	10000
7.	Маркетолог	1	9000
8.	Робітники на виробництво	2	10000
	Разом:	9	84000

Враховуючи те, що на самому початку виробництва кількість робочого персоналу не дуже велика, тому обов'язки керівника, бухгалтера, інженера та маркетолога будуть виконуватися однією людиною. В подальшому при стабільному розвитку стартап-проекту буде проводитися розширення персоналу.

Далі необхідно визначити початкові витрати необхідні для стартапу.

Таблиця 4.27. План витрат на запуск виробництва продукції

<i>№</i>	<i>Найменування</i>	<i>Характеристика</i>	<i>Вартість, тис. грн.</i>
1.	Закупівля обладнання та устаткування	Витрати на придбання обладнання	70000
2.	Сировина, основні матеріали та комплектуючі	Вартість сировини та матеріалів для забезпечення технологічного процесу	1400
4.	Паливо та електроенергія необхідна на технологічні цілі	Витрати на електроенергію, а також на паливо, необхідні для запуску проектної потужності	15000
5.	Оплата праці персоналу	Витрати на заробітну плату	84000
6.	Освоєння та запуск виробництва	Витрати на пускові роботи, запуск виробництва	20000
	Разом		190400

В даній таблиці було визначено початковий план витрат, необхідний для запуску виробництва.

Визначимо загальні обсяги інвестицій

Таблиця 4.28. Обсяги інвестицій

<i>№</i>	<i>Вид інвестицій</i>	<i>Сума, грн</i>
1.	Першо-початкові інвестиції	190400
2.	Інвестиції на стадії виробництва	100000
3.	Інвестиції на просування стартапу	40000
	Разом:	330400

В даному підрозділі було визначено загальні обсяги інвестицій необхідних для запуску стартап-проекту, в які входять ціни на сировину, матеріали, комплектуючі, заробітна плата працівникам, просування та першопочаткові інвестиції.

#### **4.6. Висновки до розділу**

Узагальнюючи проведений аналіз стартап проекту під назвою «Автоматизований портативний прилад вимірювання напруженості електричного на місцевості» можна зробити висновок, що ринкова комерціалізація проекту опирається на наявний попит, та рентабельність роботи на ринку. Беручи до уваги потенційні групи користувачів та клієнтів, перспектива впровадження проекту є достатньо високою, також цьому сприяє, відсутність прямих конкурентів, так як даний прилад має більший функціонал, це не просто вимірювач, які вже представлені на ринку, а ціла автоматизована система, аналогів якій поки не існує. Також цьому сприяє наявність власного ПЗ для відображення результатів вимірювання.

На даний момент система є унікальна, але в майбутньому можуть з'явитись конкуренти, і щоб даний прилад був на високій позиції в рейтингу в користувачів, потрібно постійно його розвивати. Можливий розвиток системи підвищення точності приладу, створення ПЗ для мобільних пристроїв, додавання систем безпроводної передачі даних, покращення інформативності виводу результатів, програмне розпізнавання аномальних зон.

Продукт відразу буде націлений на міжнародний ринок, так як можливих споживачів в середині країни, невелика кількість. А на міжнародному їх кількість велика, а товарів аналогів не існує.

Також передбачений захист на даний прилад в вигляді, реєстрації авторських прав, а також створення патентів на власні технології, що використовуються в даному патенті.

В даному розділі також був виконаний розрахунок інвестицій, для запуску даного стартап-проекту.

## **Висновки**

В даній роботі були розглянуті існуючі методи для вимірювання напруженості електричного поля. Проведений аналіз динамічного методу вимірювання, так як на даний час, цей спосіб являється найкращим, також розроблений спосіб покращення даного методу вимірювання, для підвищення точності вимірювань, при цьому не ускладнюючи сам вимірювач. Розроблено саму автоматизовану систему, на базі додавання до вимірювача технологій GPS та SD-card.

Також було проведено роботу по розробці ПЗ, для розшифрування даних вимірювання, і подачі результату в зручному для подальшого аналізу методу. Саме ПЗ можна покращити, зробивши прототипи ПЗ для систем Android і IOS, а також додавши спеціальні методи аналізу, які самі зможуть виявляти ділянки з аномальними значеннями.

Дана систему буде корисною в сферах знаходження корисних копалин, для первинного розслідування, що дозволить швидко виявляти, чи є на даній ділянці корисні копалини, а з отриманої карти місцевості, що буде результатом роботи нашого ПЗ, можна буде швидко виділити контури ділянки, на якій є корисні копалини.

Також дана система буде корисна в наукових дослідженнях явищ, на які може впливати електричне поле атмосфери, та знаходження причин появи, високих значень, або низьких значень даних полів.

## Список використаних джерел

[1]. Патент України на винахід №105423 від 12.05.2014 МПК G01R29/12 (2006.01). Бюл. №09/2014 номер заявки а201211682 Цифровий спосіб вимірювання напруженості електричного поля / Баженов В.Г. , Якимчук М.А., Грузін С.В.

[2]. Цифровий спосіб вимірювання напруженості електричного поля та іонних струмів Патент України на винахід №112812 від 25.10.2016, МПК G01K29/12 Бюл №20 /Баженов В.Г., Грузін С.В.

[3]. Имянитов И. М., Приборы и методы для изучения электричества атмосферы, М., 1957.

[4]. Метод и аппаратура для измерения напряжённости электрических полей при геолого-геофизических исследованиях / В.Г. Баженов, Н.А. Якимчук, С.В. Грузин, И.С. Пидлисна // Теоретичні та прикладні аспекти геоінформатики: Зб. наук. пр. — 2014. — Вип. 11. — С. 17-29. — Бібліогр.: 11 назв. — рос.

[5]. Bazhenov V. Mathematical Model Electro Capacitive Methods of Nondestructive Testing / Ivitska D.K. Bazhenov V.G., Ivitskiy I.I. // SWorld Journal, 2017. – Vol. 13 - pp. 185-189

[6]. А.с. 1493964 СССР МКИ G01 29/08 «Способ измерения параметров электромагнитного поля» / В.Г. Болдырев, В.В Бочаров, В.Л. Булеков, В. Л. Головкин, А.А Ковзан, С.Б. Резников, В.В. Савостьянов / №4281857/24-09 Заявлено 13.07.87 Опубликовано 15.07.89 Бюл. №26

[7]. А.с. 319905 СССР МКИ G01 19/10 «Способ исследования электрического поля» / С.И. Павлов - №1378348/26-25 Заявлено 24.11.1969 Опубликовано 02.11.1971 Бюл. №33

[8]. Бирюков С. В. Физические основы измерения параметров электрических полей/ Бирюков С.В Омск СибАДИ 2008 -112с.

[9]. А.с. 1288630 СССР МКИ G01 29/12 «Способ измерения электростатического поля» / К.С. Жупахин, В.С. Жупахин, Ю.П. Михайловский №3789010/24-21 Заявлено 05.09.1984 Опубликовано 07.02.1987 Бюл. №5



[10]. А.с. 1257567 СССР МКИ G01 29/12 «Способ измерения электростатического поля» / К.С. Жупахин - №3786094/24-21 Заявлено 28.08.1984 Опубликовано 15.09.1986 Бюл. №34

[11]. В. С. Аксельрод, В. А. Мондрусов, К. Б. Щигловский, Методы и средства измерения напряженности и потенциала электрического поля

[12]. N O Renno, J F Kok, H Kirkham, and S Rogacki, A miniature sensor for electrical field measurements in dusty planetary atmospheres, journal of Physics: Conference Series 142 (2008) 012075

[13]. А.с. 1509758 СССР МКИ G01 29/12 «Измеритель напряженности статических и квазистатических электрических полей» / Ю.Г. Пехтерев, В.В. Канюшкин, В.А. Кочнев - №4306040/24-09 Заявлено 14.09.1987 Опубликовано 23.09.1989 Бюл. №176

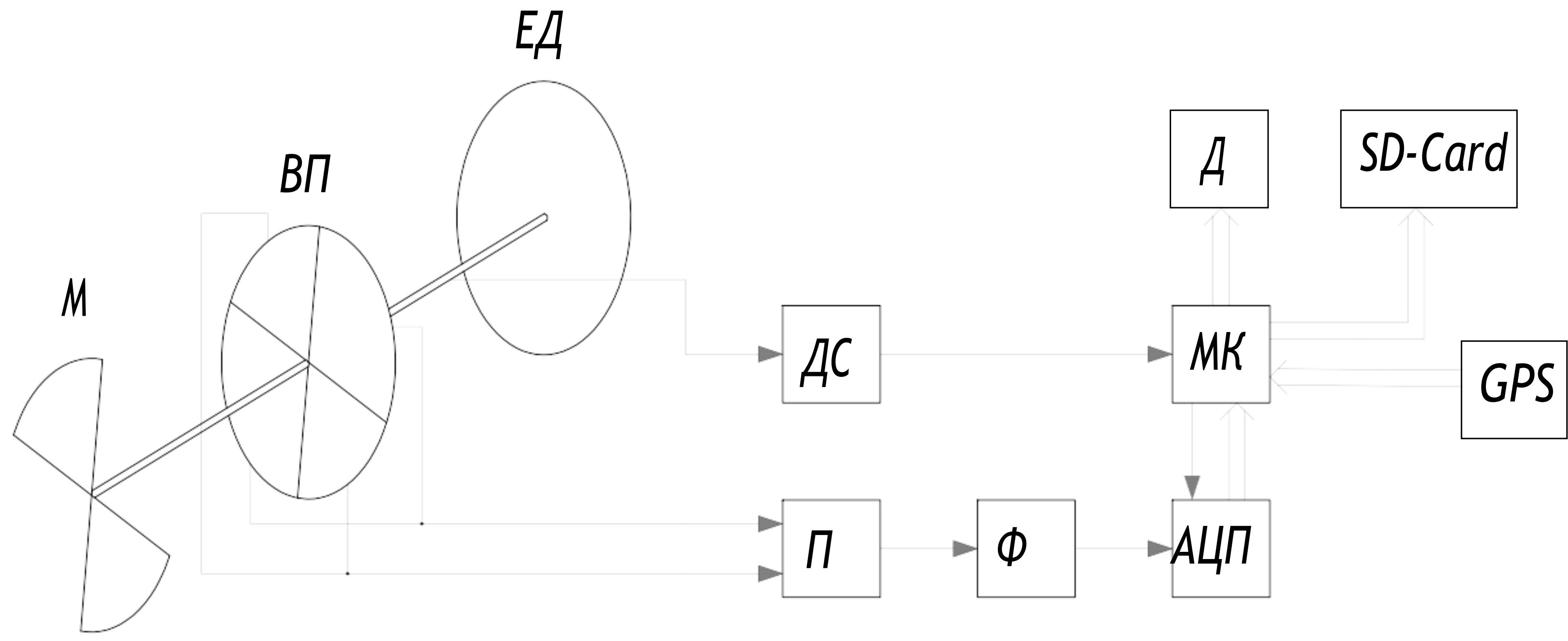
[14]. Технології електромагнітного неруйнівного контролю. Лабораторний практикум [Електронний ресурс] : навчальний посібник / Ю. В. Куц, Ю. Ю. Лисенко ; КПІ ім. Ігоря Сікорського. – Київ : КПІ ім. Ігоря Сікорського, 2021. – 63 с.

[15]. V. Bazhenov, A. Protasov, K. Gloinik, “Increasing of operation speed of digital eddy current defectoscopes based on frequency synthesizer”, in 2017 IEEE Microwaves, Radar and Remote Sensing Symposium (MRRS), 2017. doi: 10.1109/mrrs.2017.8075051

[16]. Y. Kalenychenko, V. Bazhenov, V. Koval, S. Ratsebarikiy, “Determination of mechanical properties of paramagnetic materials by multi-frequency method” in Scientific proceedings NDT days 2019 XXXI International Conference «Defectosopia 19», june 17-20. Sozopol, Bulgaria, 2019, pp.406-417

# Додатки

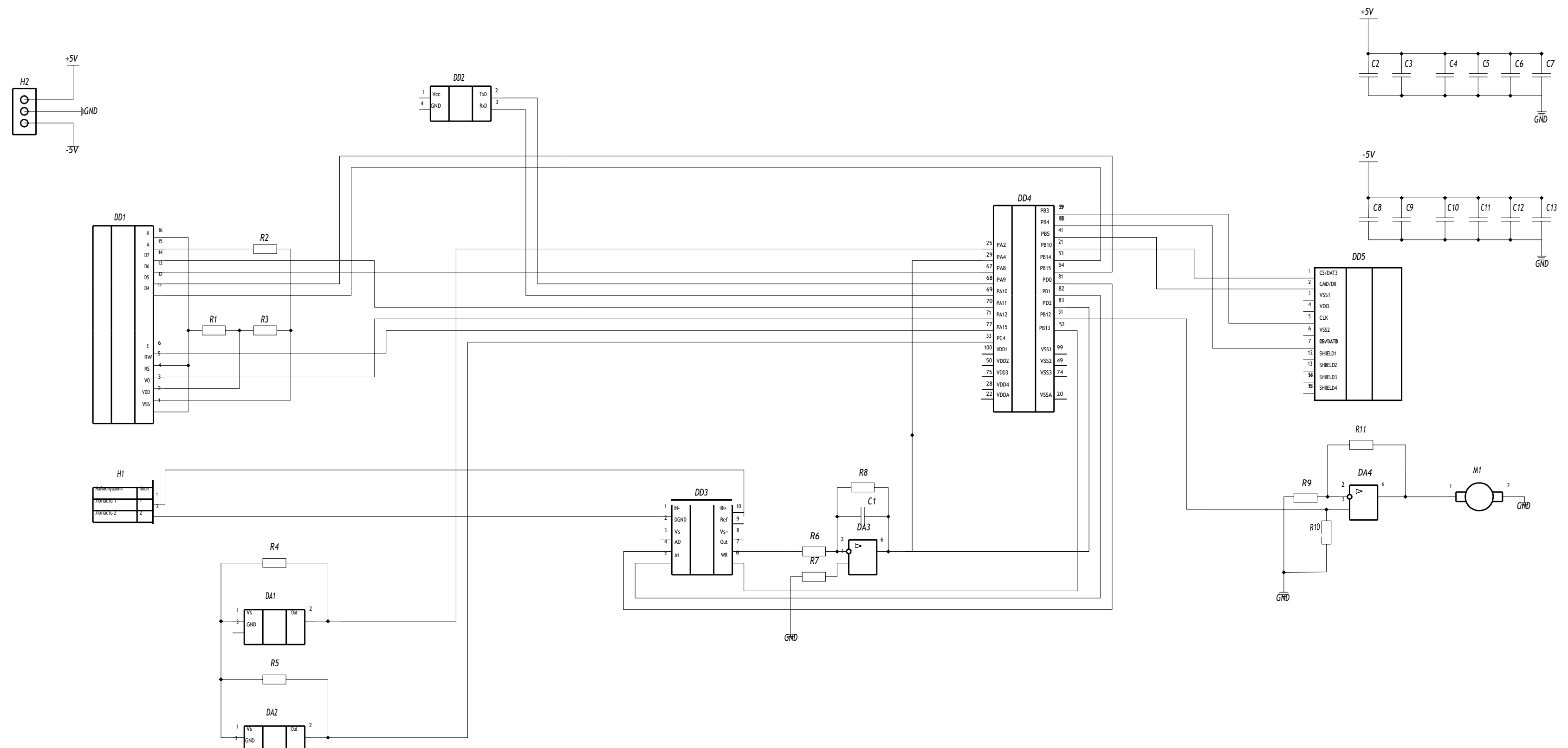
Програмний код надається по запиту  
до авторів



- М - модулятор
- ВП - вимірювальні пластини
- ЕД - електродвигун
- П - підсилювач
- Ф - фільтр
- АЦП - аналого-цифровий перетворювач
- ДС - датчик синхронізації положення пластин
- МК - мікроконтролер
- Д - дисплеї
- SD-Card - модуль пам'яті
- GPS - модуль точного позиціонування на карті

Місце № подля  
Листів у даній  
Взаєм. шифр №  
Місце № дублі  
Листів у даній  
Сторона №  
Перев. приміт.

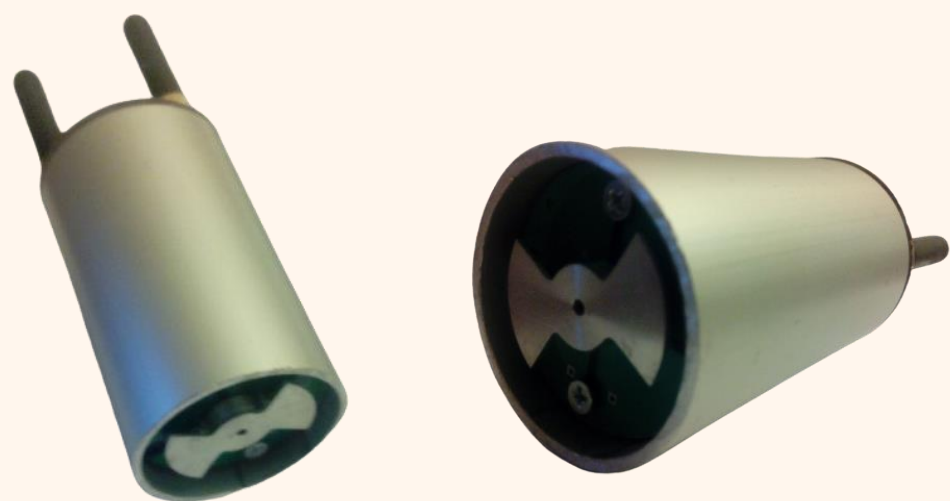
ПК01мн.0100.000 E1							Лист	Масштаб
Ізм. Лист	№ докум.	Попр.	Дата	Автоматизований Вимірювач напруженості електричного поля атмосфери на місцевості				1:1
Разроб.	Белак АС			Схема електрична структурна			Лист	Листов 1
Пров.								
Т.контр.								
Н.контр.								
Утв.	Баженев В.Г.							



Позначення мікросхеми	Виходи мікросхем, до яких необхідно підключити живлення		
	+5 В	GND	-5В
DA1, DA2	1	3	
DA3, DA4	7		4
DD1	2	1	
DD2	1	4	
DD3	8	2,9	3
DD4	22, 28, 50, 78, 100	20, 49, 74, 99	
DD5	4	3, 6, 12, 13, 14, 15	
M1		2	

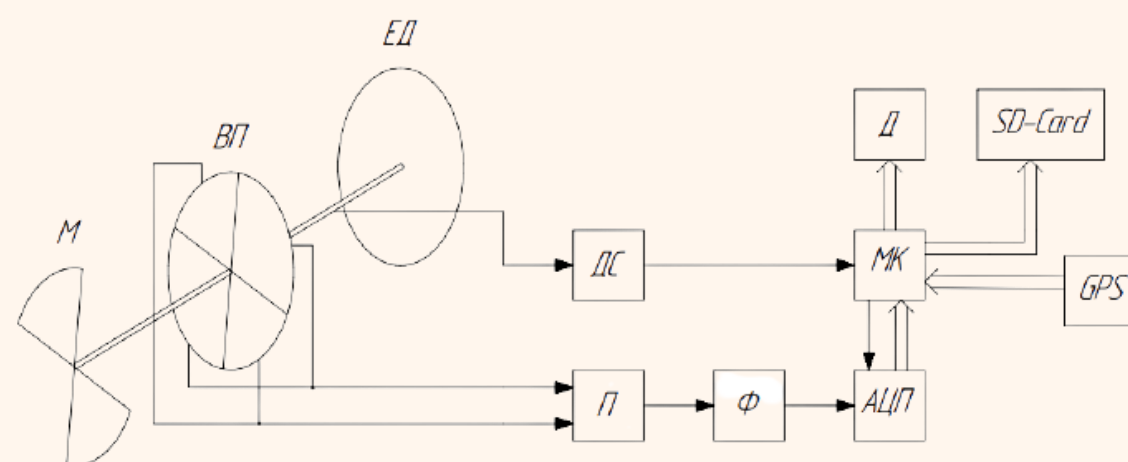
Лист № докум. Перв. примірник.  
 Сторона №  
 Листів у даній  
 Листів № змін  
 Взам. інв. №  
 Інв. № змін  
 Листів у даній

# Автоматизований прилад вимірювання напруженості електричного поля на місцевості



Ідея даної роботи полягає в розробці автоматизованої системи для вимірювання напруженості електричного поля атмосфери на місцевості. Автоматизована система складається з вимірювача електричного поля малих значень, для реєстрації змін електричного поля атмосфери, мікроконтролера, системи GPS. та SD-CARD reader. Вимірювач виконаний на базі динамічного методу, із-за його швидкодії, та точності вимірювання. Дана система являється актуальною на даний час, так як за допомогою даних вимірювань можна проводити пошук корисних копалин, а також проводити наукові дослідження, щодо аномалій напруженості електричного поля атмосфери, а також причин їх появи

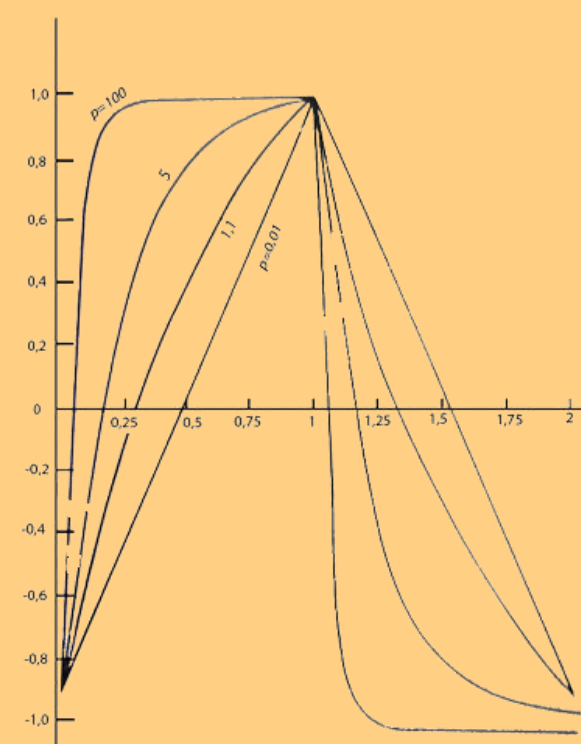
Система працює по принципу: при початку вимірювання система фіксує координати початкової точки, та запам'ятовує їх, і вимірює значення напруженості в даній точці. Після чого ці координати разом зі значенням напруженості заносяться в файл, що знаходиться на SD картці. Далі система буде постійно приймати координати, місцезнаходження системи, і вираховувати дистанцію до точки яку ми запам'ятали, якщо дистанція буде рівна, або більша ніж крок виміру, система запам'ятає нову точку, і виміряє напруженість в цій точці, після чого ці дані будуть записані в файл та SD картці. Дана процедура буде відбуватись до зупинки вимірювання.



Також для повної незалежності приладу було розроблено ПЗ для розшифрування результатів вимірювання, та зручної подачі на карті місцевості. Програмне забезпечення виконане на мові програмування C#.

Інноваційність системи полягає в :

- Автоматизованості;
- Малій вазі;
- Малому енергоспоживанні;
- Використанні провідних мікроконтролерів
- Покращений алгоритм вимірювання, що дозволив підвищити точність, та зменшити енергоспоживання
- Власне програмне забезпечення, для розшифрування даних вимірювання, та представлення їх в зручному форматі



Розроблено: Белаком Артуром Сергійовичом  
Науковий керівник: Баженов Віктор Григорович