

Міністерство освіти і науки України

Луцький національний технічний університет

(повне найменування закладу вищої освіти)

Факультет комп'ютерних та інформаційних технологій

(повне найменування факультету)

Кафедра комп'ютерної інженерії та кібербезпеки

(повне найменування кафедри)

КВАЛІФІКАЦІЙНА РОБОТА  
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»  
СИМУЛЯТОР ПОГОДНИХ УМОВ ІГРОВОГО ПРОЦЕСУ  
GAMEPLAY WEATHER SIMULATOR

спеціальність 123 Комп'ютерна інженерія

(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія

(назва освітньої програми)

Виконав: здобувач вищої освіти  
групи КІ-41  
Киндюк Максим Русланович

(підпис)

Керівник:  
к.т.н., доцент  
Костючко Сергій Миколайович

(підпис)

Кваліфікаційну роботу  
допущено до захисту  
« 08 » червня 2024 р.  
Гарант освітньої програми:  
к.т.н., доцент  
Лавренчук Світлана Василівна

(підпис)

Луцьк – 2024 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра комп'ютерної інженерії та кібербезпеки

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 123 Комп'ютерна інженерія

Освітня програма: «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

проф. Н.Черняшук

« 10 » 01 2024 р.

ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

*Киндюк Максим Русланович*

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи *Симулятор погодних умов ігрового процесу*

Керівник роботи *к.т.н., доцент Костючко Сергій Миколайович*

затверджені наказом закладу вищої освіти від «30» грудня 2023 року № 459/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи *11.06.2024р.*

3. Вихідні дані до роботи *Джерелом розробки є науково-технічна література та публікації в періодичних виданнях з даного питання, опубліковані зарубіжні та вітчизняні роботи в даній області та різні інтернет-ресурси технічного спрямування*

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):

*Вступ*

*Огляд і аналіз предметної області*

*Аналіз засобів та середовищ програмування для реалізації проекту*

*Реалізація симулятора погодних умов на основі мікроконтролера Arduino*

*Висновки*

5. Перелік графічного (ілюстративного) матеріалу:

*Імітаційна модель спалаху блискавки*

*Імітаційна модель сонячного світла*

*Імітаційна модель ефекту землетрусу*

*Імітаційна модель звуку грому*

## 6. Консультанти розділів роботи

| Розділ   | Прізвище, ініціали та посада консультанта | Підпис         |                  |
|--|---|----------------|------------------|
|  |   | завдання видав | завдання прийняв |
| <i>Аналіз проблеми за темою роботи та постановка завдань дослідження</i> | <i>Костючко С.М, доцент</i>               |                |                  |
| <i>Теоретичне дослідження та практична реалізація</i>                    | <i>Костючко С.М, доцент</i>               |                |                  |
| <i>Практична реалізація CRO-системи</i>                                  | <i>Костючко С.М, доцент</i>               |                |                  |
| <i>Нормоконтроль</i>   | <i>Багнюк Н.В., доцент</i>                |                |                  |
| <i>Гарант ОП</i>   | <i>Лавренчук С.В., доцент</i>             |                |                  |
| <i>Показник запозичень тексту</i>  |   | _____%         |                  |
| <i>Академічна доброчесність</i>  | <i>Міскевич О.І., асистент</i>            |                |                  |

7. Дата видачі завдання 10.01.2024 р.

## КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів кваліфікаційної роботи                                       | Строк виконання етапів роботи | Примітка |
|-------|---|-------------------------------|----------|
| 1.    | <i>Провести огляд літератури із досліджуваної проблеми</i>                | до 15.02.2024 р.              | Виконано |
| 2.    | <i>Провести аналіз практичного застосування симуляторів погодних умов</i> | до 15.03.2024 р.              | Виконано |
| 3.    | <i>Розробка та впровадження симулятора погодних умов</i>                  | до 04.05.2024 р.              | Виконано |
| 4.    | <i>Формування висновків</i>   | до 07.05.2025 р.              | Виконано |
| 5.    | <i>Формування списку використаних джерел</i>                              | до 10.05.2024 р.              | Виконано |
| 6.    | <i>Формування додатків</i>  | до 15.05.2024 р.              | Виконано |
| 7.    | <i>Оформлення ілюстративного матеріалу</i>                                | до 20.05.2024 р.              | Виконано |
| 8.    | <i>Нормоконтроль</i>  | до 01.06.2024 р.              | Виконано |
| 9.    | <i>Інструментальна перевірка на академічний плагіат</i>                   | до 04.06.2024 р.              | Виконано |
| 10.   | <i>Представлення кваліфікаційної роботи бакалавра до захисту</i>          | до 11.06.2024 р.              | Виконано |

Здобувач вищої освіти

\_\_\_\_\_  
(підпис)

Киндюк М.Р.

\_\_\_\_\_  
(прізвище, ініціали)

Керівник кваліфікаційної роботи

\_\_\_\_\_  
(підпис)

Костючко С.М.

\_\_\_\_\_  
(прізвище, ініціали)

## АНОТАЦІЯ

Киндюк М.Р. Симулятор погодних умов ігрового процесу. Рукопис.

Кваліфікаційна робота бакалавра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2024. 55 с.

Кваліфікаційна робота складається з вступу, трьох розділів, висновків, списку використаних джерел.

Перший розділ присвячено огляду предметної області, а саме визначенню поняття симуляції погодних умов та його класифікації. Сформовано історичні чинники, що сприяли розвитку технології та ті, що сприятимуть в майбутньому. Визначено області застосування симуляторів, проаналізовано популярні платформи та інструменти для емуляції.

В другому розділі здійснено вибір та обґрунтування вибору мікроконтролера для розробки симулятора погодних умов. Детально проаналізовано всі складові пристрою, наведено графічний матеріал. Проведено аналіз найпопулярніших середовищ програмування для мікроконтролерів Arduino, при їх порівнянні було виокремлено переваги та недоліки.

Третій розділ присвячено опису реалізації симулятора погодних умов. Було обґрунтовано вибір необхідних компонентів для реалізації кожної з задач. Створено код, проведено тестування та задокументовано результати.

Ключові слова: Arduino, симулятор погодних умов, UNO R3, мікроконтролер, кліматичні явища.

## ANNOTATION

Kyndiuk M.R. Gameplay weather simulator. Manuscript.

Qualifying work of a bachelor of EP «Computer Engineering» specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2024. 55 p.

Qualification work consists of an introduction, three chapters, conclusions and a references.

The first chapter is devoted to an overview of the subject area, namely the definition of the concept of weather simulation and its classification. Historical factors that have contributed to the development of the technology and those that will contribute in the future are identified. Areas of application of simulators are determined, popular platforms and tools for emulation are analyzed.

The second chapter makes a selection and justification of the microcontroller for the development of a weather conditions simulator. All components of the device are thoroughly analyzed and graphical material is provided. An analysis of the most popular programming environments for Arduino microcontrollers is conducted, comparing their advantages and disadvantages.

The third chapter describes the implementation of the weather conditions simulator, justifying the selection of necessary components for the implementation of each task. Creating and testing of the code is carried out, the results are documented.

Keywords: Arduino, gameplay weather simulator, UNO R3, microcontroller, climate conditions.

## ЗМІСТ

|  |    |
|--|----|
| ВСТУП.....   | 7  |
| РОЗДІЛ 1 ОГЛЯД І АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....                                     | 8  |
| 1.1 Визначення та класифікація симуляцій погодних умов.....                          | 8  |
| 1.2 Історія симуляції погодних умов.....   | 9  |
| 1.3 Области застосування .....   | 11 |
| 1.4 Огляд популярних платформ та інструментів для симуляції погодних умов.....       | 14 |
| РОЗДІЛ 2 АНАЛІЗ ЗАСОБІВ ТА СЕРЕДОВИЩ ПРОГРАМУВАННЯ ДЛЯ РЕАЛІЗАЦІЇ ПРОЕКТУ .....      | 17 |
| 2.1 Фактори вибору засобів реалізації проекту .....                                  | 17 |
| 2.2 Аналіз обраної плати.....  | 18 |
| 2.3 Вибір середовища програмування.....  | 25 |
| РОЗДІЛ 3 РЕАЛІЗАЦІЯ СИМУЛЯТОРА ПОГОДНИХ УМОВ НА ОСНОВІ МІКРОКОНТРОЛЕРА ARDUINO ..... | 28 |
| 3.1 Вступ до практичного розділу .....   | 28 |
| 3.2 Імітація спалаху блискавки.....  | 29 |
| 3.3 Симуляція сонячного світла .....   | 32 |
| 3.4 Симуляція ефекту землетрусу .....  | 36 |
| 3.5 Симуляція ефекту вітру .....   | 39 |
| 3.6 Симуляція звуку грому .....  | 41 |
| ВИСНОВКИ .....   | 45 |
| ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ .....  | 46 |
| ДОДАТКИ .....  | 50 |

## ВСТУП

Актуальність теми зумовлена тим, що в умовах стрімкого розвитку технологій і зростаючого попиту на реалістичні віртуальні середовища, розробка симуляторів, що можуть відтворювати різноманітні погодні явища набуває все більшого значення. Використання такого симулятора може сприяти розвитку наступних сфер діяльності: наукові дослідження, освіта, технології Інтернету речей, зростанню індустрії відеоігор, тощо.

Метою роботи є розробка та реалізація пристрою, здатного симулювати погодні умови у середовищі програмування Arduino.

Об'єкт дослідження – система симуляції погодних умов, що реалізована у середовищі програмування Arduino.

Предмет дослідження – методи й засоби створення інтерактивного симулятора погодних умов.

Завдання, які необхідно виконати:

- Провести аналіз сучасних технологій та рішень для симуляції погодних умов.
- Обґрунтувати вибір електронних компонентів, модулів та середовища програмування, необхідних для реалізації проекту.
- Розробити програмне забезпечення для керування компонентами згідно з обраними погодними умовами.
- Провести тестування й налагодження системи.

## РОЗДІЛ 1

### ОГЛЯД І АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

#### 1.1 Визначення та класифікація симуляцій погодних умов

Симуляція погодних умов – це процес створення віртуальної моделі реальних погодних явищ, таких як дощ, сонячне світло, вітер, сніг, температура навколишнього середовища, тощо. Метою такої симуляції є створення умов, які максимально наближені до реальних погодних явищ, для їх подальшого вивчення, тестування або використання в різних прикладних сферах. Симуляція погодних умов може бути реалізована двома способами: програмно та апаратно. Програмна симуляція включає в себе наступні аспекти:

- Використання складних математичних моделей для опису та прогнозування поведінки атмосферних систем. Це включає в себе моделювання динаміки рідин, термодинамічних процесів, тощо [1].

- Розробка алгоритмів, що дозволяють створювати віртуальні погодні умови в комп'ютерних програмах. Такі алгоритми можуть включати генерацію випадкових чисел для створення різноманітних погодних сценаріїв, симуляцію впливу вітру на різні об'єкти, тощо.

- Використання графічних технологій для візуалізації погодних явищ у відеоіграх, тренажерах, навчальних програмах та інших додатках. Це може включати рендеринг дощу, снігу, туману, зміни освітлення та інших ефектів [2].

Апаратна ж симуляція для імітації погодних умов використовує безпосередньо фізичні пристрої, як, наприклад, лампи, вентилятори, нагрівачі, тощо. Наприклад, лампа може імітувати сонячне світло, вентилятор – вітер, а нагрівач – зміну температурних умов. Нерідко при апаратній симуляції застосовуються мікроконтролери (наприклад, Arduino), за допомогою яких здійснюється управління електронними компонентами (рис. 1.1).

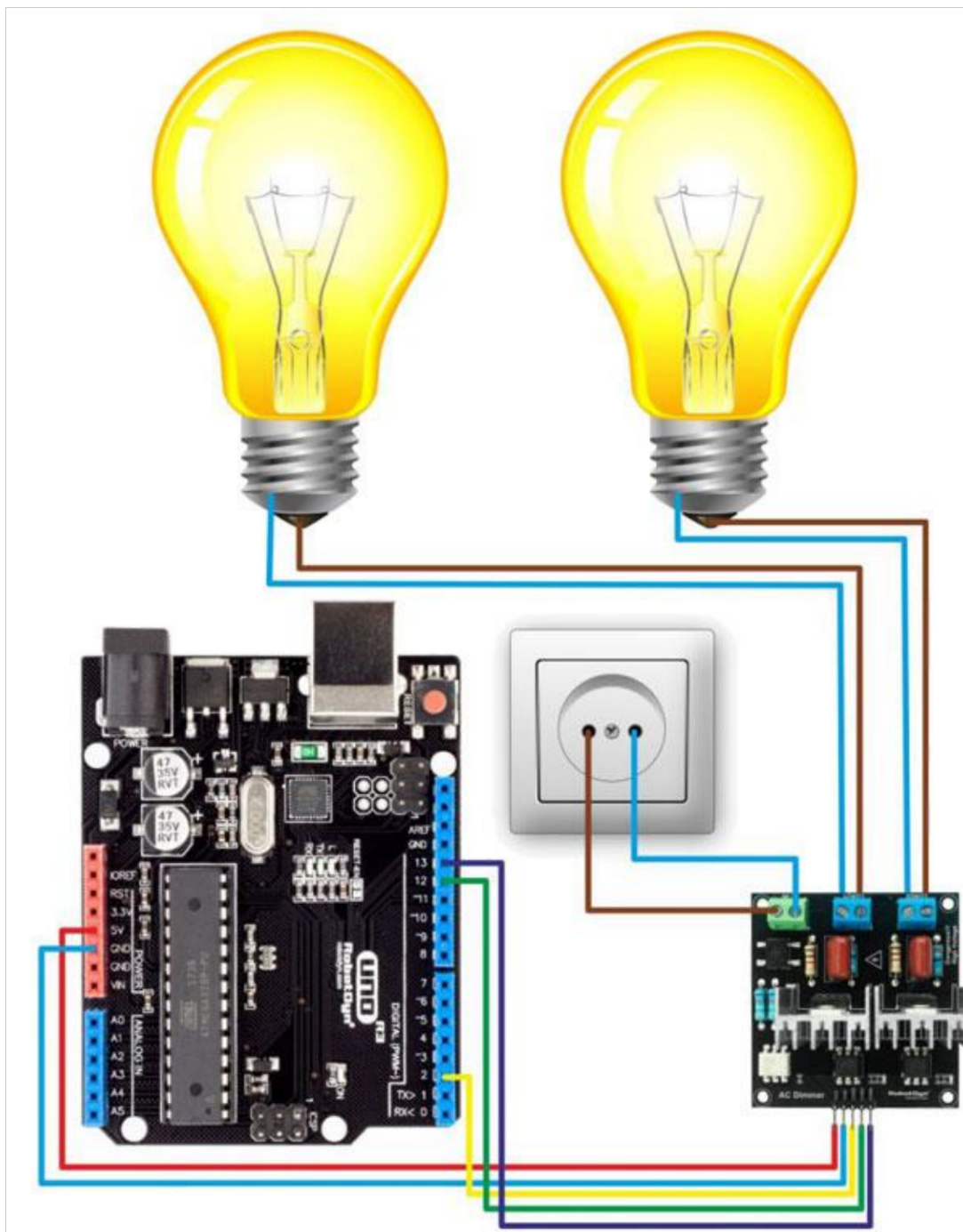


Рисунок 1.1 – Приклад використання Arduino мікроконтролера [21]

## 1.2 Історія симуляції погодних умов

Історія симуляції погодних умов бере свій початок ще з давніх часів, коли люди намагались передбачити погоду за допомогою спостережень і примітивних інструментів. Ранні спроби прогнозування погоди включали використання календарів, астрономічних явищ та народних прикмет, однак справжні наукові методи моделювання почали розвиватись значно пізніше.

Протягом XVIII-XIX століть науковці почали розробляти перші математичні моделі атмосфери, а дослідження з динаміки рідин, такі як рівняння Бернуллі [3], заклали основу для розуміння процесів та поведінки, що притаманні атмосферним явищам [4].

У 1827 році Жан Батист Жозеф Фур'є зробив значний внесок у розвиток кліматології та метеорології своєю теорією теплопровідності. Його роботи заклали основу для розуміння теплових процесів в атмосфері [5].

На початку XX століття, більш складні математичні моделі та чисельні методи для розв'язання рівнянь, що описують погодні явища, стали можливі завдяки розвитку комп'ютерних технологій. У 1930-их роках навіть була запропонована ідея використання чисельних методів для прогнозу погоди, однак тодішні обчислювальні потужності були недосатніми для реалізації такого підходу [6].

У 1960-их роках, коли вже почали з'являтися більш потужні комп'ютери, стала можливою реалізація чисельних методів для моделювання атмосфери, проте перші комерційні програми, що дозволяли симулювати погодні умови, вийшли на ринок лише у 1980-их роках. Вони використовувались в різноманітних галузях, що включали авіацію, судноплавство, сільське господарство, тощо. Розвиток графічних технологій дозволив створювати реалістичні візуалізації погодних умов.

Початок XXI століття ознаменував собою початок епохи активного використання Інтернету речей (IoT) та розумних пристроїв для моніторингу та симуляцій. Розробка платформ на базі Arduino та інших мікроконтролерів дозволила створювати інтерактивні симулятори погодних умов для освітніх і наукових цілей.

Станом на сьогодні, для створення високоточних моделей клімату використовуються суперкомп'ютери, а інтеграція симуляцій погодних умов у віртуальну (VR) та доповнену (AR) реальність стала невід'ємною складовою навчальних та науково дослідницьких середовищ.

Зважаючи на все це, можна припустити, що з надзвичайно стрімким розвитком штучного інтелекту (AI) й нейромереж у майбутньому роль моніторингу, прогнозування та симуляцій погодних умов буде тільки зростати, що тільки підтверджує високу актуальність теми. Отже, історія симуляції погодних умов

демонструє явну еволюцію від простих спостережень до складних математичних моделей. З розвитком комп'ютерних технологій, імітація стала важливим інструментом у різних галузях, від освіти та науки до індустрії розваг та інженерії, а його майбутнє обіцяє ще більш щільну інтеграцію з передовими технологіями задля створення ще реалістичніших та інтерактивніших моделей клімату.

### 1.3 Області застосування

Симуляція погодних умов є важливим і багатоцільовим інструментом, що знаходить застосування у різних галузях завдяки можливостям сучасних технологій, що дозволяють створювати реалістичні і контрольовані середовища для аналізу, навчання і оптимізації різних процесів. Приклади сфер, які отримують з цього значні переваги включають в себе наступні:

– Відеоігри та розваги, де симуляція погодних умов відіграє важливу роль у створенні реалістичних та захоплюючих віртуальних світів (рис. 1.2). Додавання динамічних ефектів, як, наприклад, дощ, сніг, туман та вітер, покращує атмосферу ігрового процесу та сприяє більш глибокому зануренню гравців [7].



Рисунок 1.2 – Симуляція погодних умов у відеоіграх [19]

– В кінематографі імітатори застосовуються з метою створення візуальних ефектів, що дозволяють досягнути необхідної атмосфери. Наприклад,

машина для створення штучного снігу (рис. 1.3) допомагає в створенні зимових пейзажів навіть улітку, коли справжнього снігу бути не може. Також це гарантує безпеку знімальної групи, адже знімати можна в студійних умовах.



Рисунок 1.3 – Машина для створення штучного снігу [10]

– Освітні процеси, де симуляція погодних умов дозволяє студентам вивчати природні явища і їх вплив на навколишнє середовище в інтерактивній формі. Віртуальні лабораторії та симулятори дозволяють проводити експерименти і спостереження, які можуть бути небезпечними для виконання в реальному світі [8].

– При проведенні наукових досліджень імітатори погодних умов дозволяють вченим моделювати і вивчати складні атмосферні процеси, як, наприклад, вивчення кліматичних змін, аналіз впливу різних факторів на погоду, вивчення довгострокових кліматичних тенденцій, тощо [9].

– Також симуляції використовують для тестування будівельних матеріалів у різних кліматичних умовах, наприклад, моделювання впливу атмосферних умов на транспортні системи, дренажні системи, водопостачання, тощо.

– При плануванні сільськогосподарських операцій симуляції погодних умов допомагають оцінювати ризики та планувати посівні й збиральні кампанії. Одним з найбільш перспективних досягнень агротехніки є використання безпілотних повітряних літальних апаратів (БПЛА) (рис. 1.4) для виконання завдань, які раніше вимагали людської праці, наприклад, обмір полів, посів покривних культур, боротьби зі шкідниками, тощо.



Рисунок 1.4 – Приклад використання БПЛА в агросфері [22]

– Імітатори використовуються і в транспортній та логістичній індустрії, в основному для планування маршрутів та забезпечення безпеки перевезень. Вони допомагають прогнозувати умови на дорогах, в повітрі та на морі, що дозволяє зменшити ризики та збільшити ефективність перевозів.

– В енергетичній галузі симуляції використовуються для прогнозування виробництва відновлювальних джерел енергії, таких, як вітрові та сонячні електростанції, а також для управління енергомережами.

– Насамкінець, симулятори погодних умов відіграють важливу роль у військових операціях та навчаннях, допомагаючи планувати дії та оцінювати вплив кліматичних змінних на бойові операції, наприклад, використання симуляторів, що дозволяють військовим тренуватись в бойових умовах. Також аналіз атмосферних показників дозволяє планувати розгортання військ, логістику та забезпечення.

## 1.4 Огляд популярних платформ та інструментів для симуляції погодних умов

Як вже було зазначено вище, симулятори погодних умов справді мають широку область використання, що охоплює різномантні сфери діяльності, тому варто оглянути, які ж популярні платформи та інструменти використовуються задля досягнення цих цілей:

– Weather Research and Forecasting Model (WRF) – потужна числова модель, розроблена для дослідження атмосферних процесів та оперативних прогнозів погоди. Серед особливостей можна виокремити високу роздільну здатність, що охоплює відстані від десятка метрів до кількох тисяч кілометрів, що дозволяє отримувати та візуалізувати точні метеорологічні дані (рис. 1.5), на основі яких створюються вже симуляції з ідеалізованими умовами, якими користуються вчені при своїх дослідженнях [11].

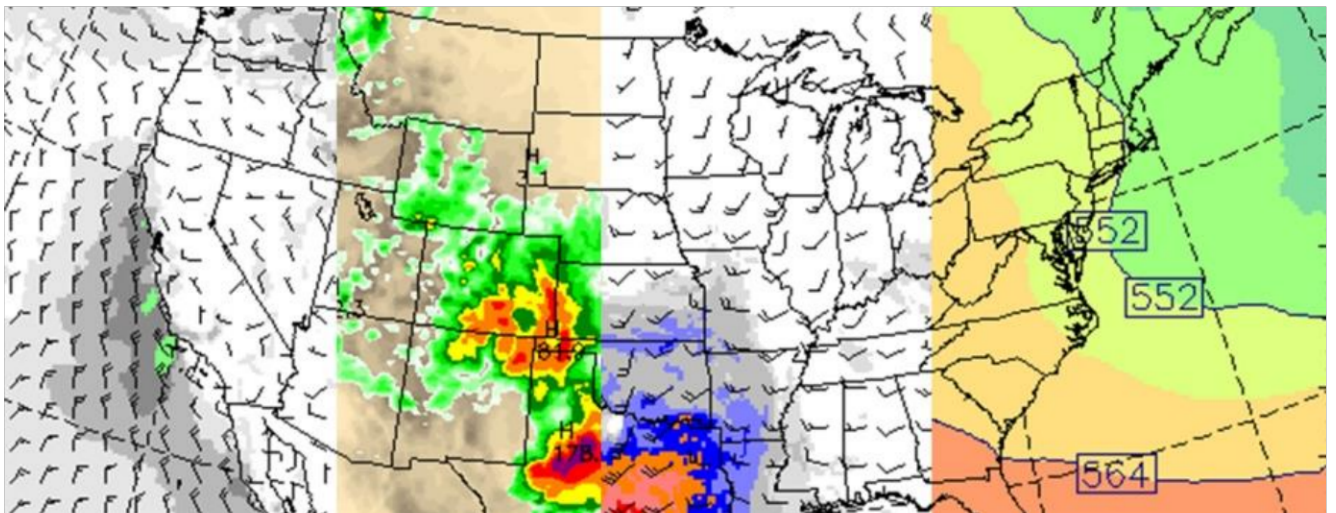


Рисунок 1.5 – Візуалізація метеорологічних даних на прикладі WRF [11]

– Global Forecast System – глобальна числова модель прогнозування погоди, розроблена Національною океанічною та атмосферною адміністрацією США. Вона є однією з найважливіших та найбільш використовуваних моделей у всьому світі, оскільки забезпечує детальні прогнози, що охоплюють всю земну кулю та оновлюються регулярно. Використовується при кліматичних

дослідженнях для аналізу довгострокових тенденцій та дослідження впливу зовнішніх факторів на клімат, та при підтримці авіаційних і морських операцій [12].

– Unity – одна з найпопулярніших платформ для розробки ігор, яка дозволяє створювати як двовимірні, так і трьохвимірні ігрові світи. Вона підтримує інтеграцію різних SDK (Software Development Kits) для інтеграції реалістичних погодних умов у ігрові проекти та віртуальні середовища, що дозволяє розробникам створювати динамічні й захоплюючі ігрові світи. Серед таких SDK можна виокремити Enviro - Sky and Weather (рис. 1.6), Weather Maker та uSkyPro.



Рисунок 1.6 – Симуляція погодних умов на основі SDK Enviro [20]

– NOAA's Environmental Modeling System (NEMS) – структура, що поєднує різні числові моделі та інструменти для симуляції погодних умов та кліматичних досліджень. Вона також розроблена Національною океанічною та атмосферною адміністрацією США з метою забезпечення більшої точності та ефективності при аналізі кліматичних змін. Серед особливостей варто виокремити можливість об'єднувати різні атмосферні, океанічні та земельні моделі в єдину структуру, забезпечуючи комплексний підхід до симуляції погодних умов.

– MeteoBlue – комерційна платформа, що надає точні прогнози та кліматичні симуляції для широкого кола користувачів. Заснована у 2006 році, MeteoBlue використовує численні метеорологічні моделі та алгоритми для створення високоточних прогнозів, що користуються популярністю в широких масах завдяки зручним та інтуїтивно зрозумілим інтерфейсам (рис. 1.7), інтегрованим в їх веб-сайт та мобільний додаток [14].

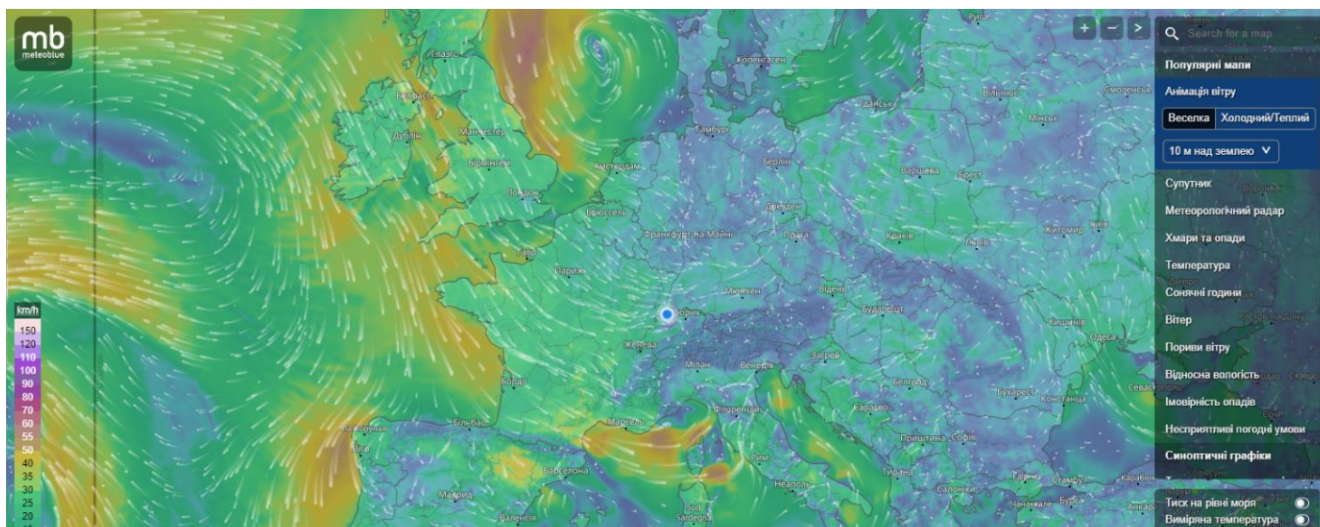


Рисунок 1.7 – Візуалізація погодніх умов від MeteoBlue [14]

Отже, сучасні інструменти для симуляції погодніх умов відіграють ключову роль у прогнозуванні погоди, кліматичних дослідженнях, розробці ігор та управлінні різними видами діяльності, що залежать від метеорологічних даних. Вони забезпечують високу точність, гнучкість ефективність та зручність у застосуванні, що робить їх невід’ємною складовою сучасної кліматології.

## РОЗДІЛ 2

### АНАЛІЗ ЗАСОБІВ ТА СЕРЕДОВИЩ ПРОГРАМУВАННЯ ДЛЯ РЕАЛІЗАЦІЇ ПРОЕКТУ

#### 2.1 Фактори вибору засобів реалізації проекту

Проект буде виконано на основі мікроконтролера, тому його вибір є критично важливим етапом у процесі розробки, адже правильно вибраний мікроконтролер може суттєво вплинути на ефективність, вартість та складність реалізації. Для створення пристроя симуляції погодних умов було розглянуто декілька важливих факторів, які вплинули на остаточний вибір:

- Функціональні можливості та продуктивність – мікроконтролер повинен мати достатню кількість входів та виходів для підключення виконавчих механізмів, що симулюють погодні умови. Оскільки завдання не потребує складних математичних обчислень, достатньо буде й базової обчислювальної потужності.

- Простота в процесі програмування – мікроконтролер повинен підтримувати популярні середовища розробки, що спростить процес написання коду й налагодження. Також важливим фактором є наявність у відкритому доступі готових бібліотек, що спростять інтеграцію з різними модулями.

- Доступність – мікроконтролер повинен бути доступним в плані вартості. Також він має бути наявним у місцевих ринках та радіоринках чи в інтернет-магазині.

- Спільнота та підтримка – важливим фактором є наявність у відкритому доступі документної бази, що допоможе швидко освоїти роботу з мікроконтролером, чи активної спільноти користувачів, здатних надавати підтримку у вирішенні проблем.

## 2.2 Аналіз обраної плати

Звертаючи увагу на вищезазначені фактори, при виборі було розглянуто кілька варіантів, як, наприклад, STM32, ESP32, PIC та інші, проте остаточний вибір був наданий в користь мікроконтролера на базі Arduino, а саме Keyestudio UNO R3 (рис. 2.1). Вибір був зумовлений тим, що обраний мікроконтролер відповідає всім заданим критеріям: він має достатню кількість входів та виходів, достатню обчислювальну потужність, є повністю сумісним з сучасними середовищами програмування пристроїв на основі Arduino, він є доступним за ціною, а в мережі Інтернет існує велика кількість ресурсів, включаючи навчальні посібники, відеоуроки та проекти.



Рисунок 2.1 – Обраний мікроконтролер [15]

По суті, цей мікроконтролер є високоякісною копією уже існуючої плати Arduino UNO третьої ревізії, проте він не використовує для підключення міст USB-UART FDTI, адже цю функцію в ньому виконує мікроконтролер ATmega16U2 (рис. 2.2), що забезпечує більшу швидкість передачі даних. Для програмування використовується спрощена версія C++, а розробку можна вести як і за допомогою будь-якого C/C++ інструментарію, так і використовуючи спеціалізовані IDE. Процес програмування відбувається під керуванням резидентного завантажчика за

протоколом STK500, апаратний програматор при цьому не потребується. Від своєї попередньої версії мікроконтролер відрізняється наявністю додаткових виходів інтерфейсу I2C, вбудованим аналого-цифровим перетворювачем, посиленням ланцюгом сигналу Reset та заміною інтерфейсного контролера з 8U2 на покращену його версію 16U2.

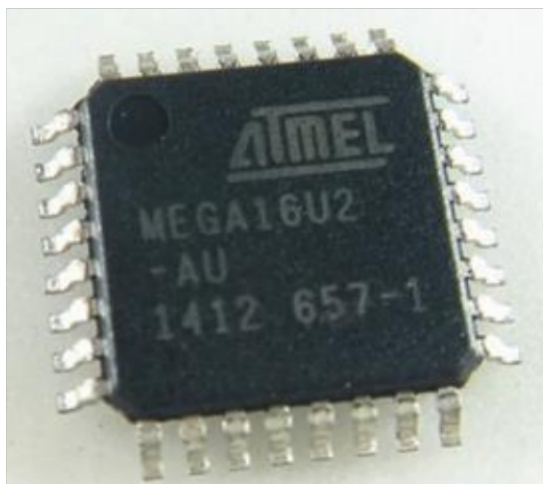


Рисунок 2.2 – Мікроконтролер для зв'язку з ПК через порт USB [24]

Серед основних технічних характеристик можна виокремити наступні:

- Мікроконтролер: ATmega328.
- Робоча напруга: 5В.
- Напруга живлення (рекомендована): 7-12В.
- Напруга живлення (гранична): 6-20В.
- Цифрові входи/виходи: 14 (6 з них можуть працювати в режимі ШІМ).
- Аналогові входи: 6.
- Сила струму на входах/виходах: 40мА.
- Сила струму для виходу 3.3В: 50мА.
- Об'єм флеш пам'яті 32 кБ (з них 0.5кБ зайнято під бутлоадер).
- SRAM: 2кБ.
- EEPROM: 1кБ.
- Робоча частота: 16МГц.
- Розмір: 68 x 53 x 15 мм.
- Вага плати: 25 г.

Arduino UNO R3 складається з мікроконтролера ATmega328, що виконує роль головного процесора, мікроконтролера ATmega16U2, що забезпечує зв'язок плати з ПК через порт USB, за допомогою якого й відбувається завантаження програм, ISCP роз'ємів для прошивки мікроконтролерів, шини живлення, шин цифрових та аналогових входів, кнопки RESET, світлодіодів живлення, передачі даних по протоколу UART, та світлодіода, підключеного до 13 контакту плати.

ATmega328 (рис. 2.3) – це головний 8-розрядний мікроконтролер плати, який виконує програмні інструкції. Він має три типи пам'яті:

- 32 Кб енергонезалежної флеш-пам'яті, що використовується для зберігання програми;
- 2 Кб енергозалежної SRAM пам'яті, що є необхідною для зберігання змінних, які використовуються програмою під час роботи;
- 1 КБ енергонезалежної EEPROM пам'яті.



Рисунок 2.3 – Основний мікроконтролер на платі [24]

З периферії (рис. 2.4) мікроконтролер має:

- 2 8-бітних лічильника зі спеціальним порівнянням каналів та регістром періодів;
- 1 16-розрядний лічильник з записом вхідних даних та каналами порівняння;

- 1 USART (універсальний синхронний/асинхронний приймач/передатчик) з генератором дробової швидкості передачі даних і визначенням початку кадру;
- 1 послідовний периферійний інтерфейс (SPI);
- 1 дворежимний периферійний контролер I2C;
- 1 аналоговий компаратор (AC) з масштабованим входом опору;
- 1 сторожовий лічильник з вбудованим генератором;
- 6 каналів ШІМ.

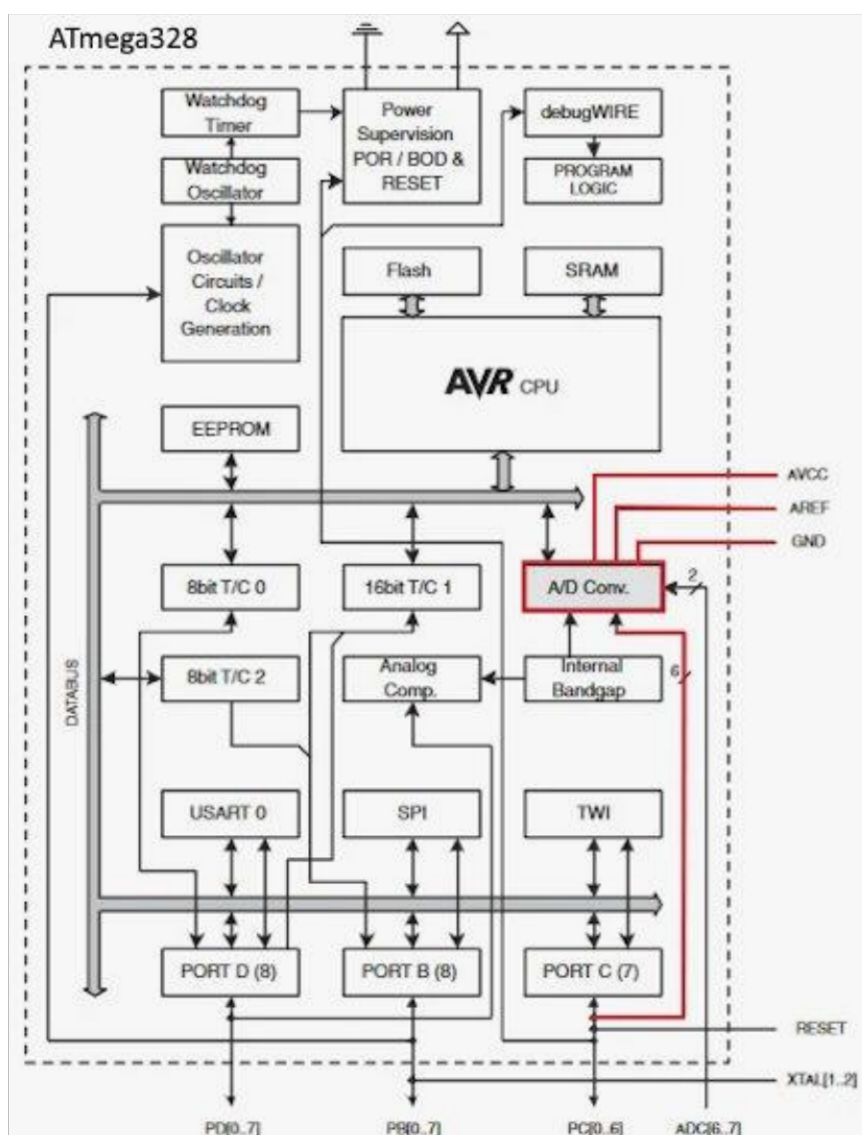


Рисунок 2.4 – Функціональна блок-схема мікроконтролера ATmega328 [23]

Щодо системи живлення: плата UNO R3 може отримувати енергію від USB порту чи зовнішнього джерела живлення, що обирається автоматично. В якості

зовнішнього джерела можна використовувати мережевий адаптер чи батарею. Адаптер підключається через роз'єм діаметром 2.1мм (центральний контакт – позитивний). Батарея ж підключається до контактів GND чи Vin роз'ємів POWER.

Напруга зовнішнього джерела може знаходитись в діапазоні від 6 до 20В, але рекомендується не допускати зниження напруги нижче 7В щоб уникнути нестабільної роботи пристрою. Також не бажано підвищувати напругу живлення понад 12В, адже стабілізатор може перегрітись та вийти з ладу.

Для підключення живлення можуть бути використані наступні виходи:

- VIN – живлення плати від зовнішнього джерела. Не зв'язане з живленням 5В від USB чи виходами інших стабілізаторів. Через цей контакт можна живити пристрої, якщо плата отримує енергію від адаптера;

- 5V – вихід стабілізатора напруги плати. На ньому напруга 5В незалежно від способу живлення. Живити плату через цей вивід не рекомендується, адже стабілізатор не буде використовуватись, що може спричинити вивід мікроконтролера з ладу;

- 3.3V – напруга 3.3В від стабілізатора напруги на платі. Максимально допустимий струм становить 50мА;

- GND – виводи «землі»;

- IOREF – надає платформам інформацію про робочу напругу мікроконтролера – 5В або 3.3В.

Кожен з 14 цифрових виходів може бути використаний як в якості входу, так і виходу. Рівень напруги на виходах становить 5В. Рекомендовано обмежувати вхідний та вихідний струм на рівні 20мА. Максимально допустиме значення параметру – 40мА. Кожен вивід має внутрішній резистор з опором 20-50кОм. Резистор може бути вимкнений програмним чином. Деякі виходи можуть виконувати додаткові функції (рис. 2.5).

- Послідовний інтерфейс: виводи 0 (Rx) та 1 (Tx). Використовуються для прийому (Rx) та віддачі (Tx) даних за послідовним інтерфейсом. Ці виводи з'єднані з відповідними виводами мікроконтролера, що виконує роль USB-UART перетворювача.



Модуль UNO R3 має також і комунікаційні інтерфейси для зв'язку з комп'ютером, іншими мікроконтролерами чи іншою платою UNO. Для цього на платі існує інтерфейс UART з логічними рівнями TTL (5В), зв'язаний з виводами 0 (Rx) та 1 (Tx). Мікросхема ATmega16U2 зв'язує цей інтерфейс із USB портом комп'ютера. При підключенні до порту, з'являється віртуальний COM порт, через який програми і працюють з платою Arduino. Прошивка мікросхеми використовує стандартні драйвери USB-COM, завантаження та встановка додаткових драйверів не потребується. Для операційної системи Windows необхідний відповідний файл з розширенням .inf. В інтегроване середовище розробки (IDE) програмного забезпечення вже включений монітор обміну по послідовному інтерфейсу, який дозволяє посилати та отримувати з плати прості текстові дані. На платі є світлодіоди RX та TX, які індують стан відповідних сигналів для зв'язку через USB (це не включає в себе послідовний інтерфейс на виводах 0 та 1).

Для того, щоб не доводилось кожен раз перед завантаженням програми натискати кнопку RESET, на платі реалізована апаратна функція скиду, що ініціюється з підключеного комп'ютера. Один з сигналів керування потоком даних (DTR) мікроконтролера ATmega16U2 підключений до виводу скидання мікроконтролера через конденсатор ємністю 0,1мкФ. Коли сигнал DTR переходить в низький стан, формується імпульс скидання мікроконтролера. Це рішення дозволяє завантажувати програму одним натисканням кнопки безпосередньо з інтегрованого середовища розробки.

Варто зауважити, що у такої функції є і свої недоліки. При підключенні плати UNO до комп'ютера, на якому встановлена операційна система Mac OS чи Linux, мікроконтролер буде скидуватись при кожному з'єднуванні програми з платою. Протягом половини секунди на платі UNO буде запущений завантажувач. Попри те, що програма ігнорує сторонні дані, вона може прийняти одразу декілька байтів з пакету одразу після встановлення підключення. Якщо в програмі на платі Arduino передбачено отримання будь-яких даних при першому запуску, необхідно відправляти їх з затримкою приблизно в 1 секунду.

На модулі UNO існує доріжка, яку можна перерізати для відключення функції автоматичного скидання. Вона маркується як «RESET-EN» (рис.2.6).

Автоматичне скидання також можна заборонити шляхом підключення резистора з опором 110Ом між лінією живлення 5В та виводом RESET.

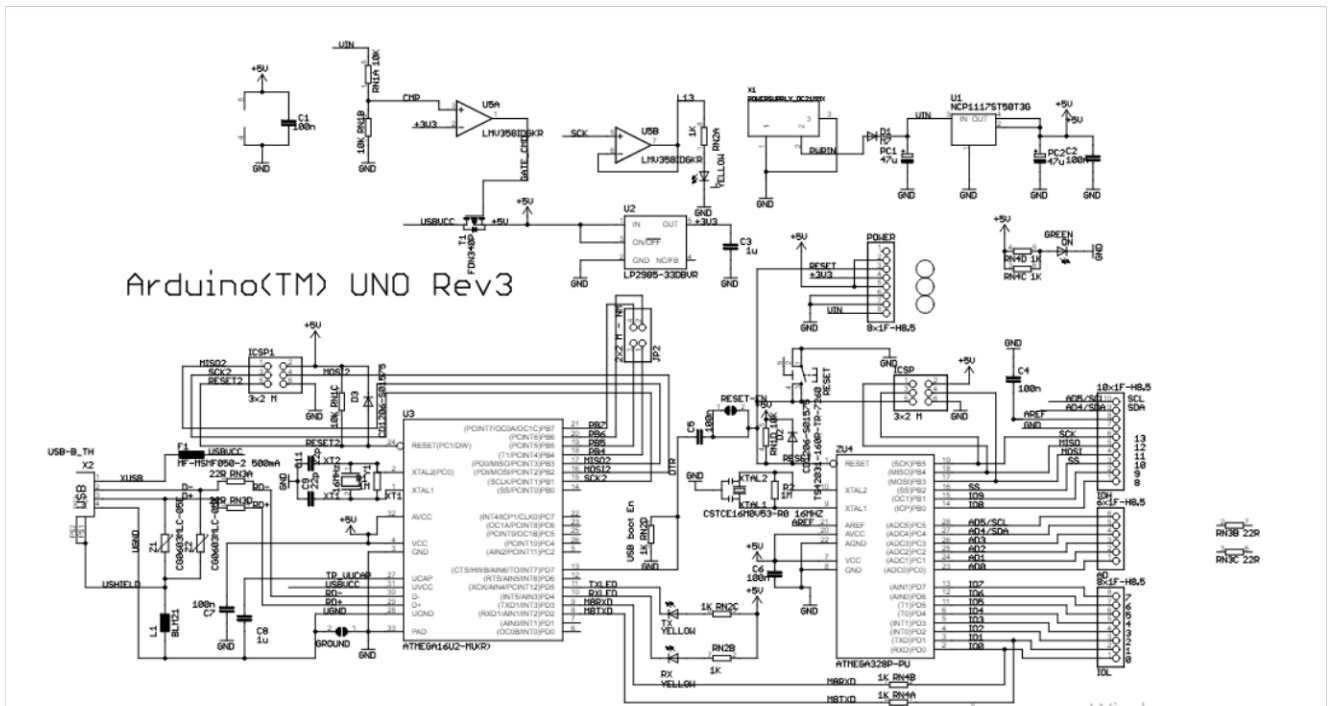


Рисунок 2.6 – Електрична схема Arduino UNO R3 [27]

### 2.3 Вибір середовища програмування

Визначивши основу проекту, варто обрати й середовище програмування, в якому і буде проводитись його реалізація. Вибір середовища програмування є важливим етапом у розробці проектів на базі мікроконтролерів, адже саме від нього залежать такі фактори, як зручність розробки, доступність бібліотек, загальна ефективність роботи над проектом та наявність інформаційної бази у відкритих джерелах. При аналізі буде розглянуто кілька популярних середовищ програмування для Arduino платформ, буде проведено їх порівняння та виокремлено основні переваги й недоліки:

- Arduino IDE є офіційним середовищем програмування для платформ Arduino. Воно забезпечує простий інтерфейс для створення, компіляції та завантаження програмного коду на плату. Серед переваг можна виділити простоту у використанні, велику кількість доступних бібліотек, активну спільноту

користувачів та повну підтримку основних операційних систем, таких, як Windows, macOS та Linux. Основним ж недоліком є обмежена можливість розширеного налагодження та управління проектами, оскільки середовище програмування не підтримує апаратне налагодження, що дозволяє програмістам встановлювати брейкпоінти, виконувати код покроково, переглядати й правити змінні прямо під час виконання коду, тощо.

– PlatformIO є потужним середовищем для розробки, що підтримує багато мікроконтролерних платформ. Серед переваг можна виокремити інтеграцію з Visual Studio Code, можливості в управлінні бібліотеками та розширений процес налагодження. Недоліком можна назвати більш складний інтерфейс, що автоматично означає підвищений поріг входження для програмістів початківців.

– Atmel Studio це інтегроване середовище розробки від компанії Atmel (нині називається Microchip), орієнтоване на AVR та SAM мікроконтролери [16]. Основною перевагою є повна інтеграція з іншим інструментарієм від Microchip, проте орієнтованість на досвідчених користувачів та обмежена підтримка бібліотек Arduino роблять цей варіант не настільки привабливим.

– Visual Micro це плагін для Visual Studio, що дозволяє використовувати можливості Visual Studio при розробці Arduino-проектів. Він включає в себе всі переваги Visual Studio, такі, як розширені налагоджувальні інструменти, та зручне управління проектами під час розробки, проте його використання при виконанні простих задач є невиправданим.

– Codebender це онлайн середовище для програмування Arduino, що дозволяє писати, компілювати та завантажувати код безпосередньо з браузера. Попри доступність з будь-якого пристрою, підключеного до мережі Інтернет, можливості при розробці є ще більш обмежені, ніж у десктопної версії.

– SimulIDE – симулятор для вивчення електроніки, зокрема Arduino, що дозволяє тестувати програми навіть якщо фізичного доступу до плати немає. Хоч можливість до симуляції та зручність у використанні і є вагомими перевагами, проте функціональність є більш обмеженою в порівнянні з популярними середовищами, а спільнота користувачів майже відсутня.

Отже, зважаючи на вищезазначені фактори, було прийняте рішення, що найбільш оптимальним середовищем розробки для створення пристрою, здатного симулювати погодні умови, буде саме Arduino IDE. Комбінація простоти у використанні, наявності всього необхідного інструментарію для ефективного створення проекту та широкої користувацької бази робить цей вибір найбільш логічним. Так, Atmel Studio та PlatformIO пропонують значно ширші можливості при налагодженні, проте складність робить їх недоцільною при розробці більш простих проектів на основі Arduino.

## РОЗДІЛ 3

### РЕАЛІЗАЦІЯ СИМУЛЯТОРА ПОГОДНІХ УМОВ НА ОСНОВІ МІКРОКОНТРОЛЕРА ARDUINO

#### 3.1 Вступ до практичного розділу

У цьому розділі розглянута безпосередньо реалізація проекту пристрою, здатного симулювати погодні умови. Симуляція включає в себе моделювання наступних програмних явищ:

- Блискавка – моделюється за допомогою світлодіода, який створює ефект спалаху.
- Сонце – представлено лампою розжарювання, яка забезпечує яскраве світло.
- Вітер – реалізується шляхом обертання кулера, що контролюється за допомогою мотору.
- Землетрус відтворюється за допомогою вібромоторчика, що при спрацюванні створює вібрації.
- Грім імітується за допомогою активного п’єзодинаміка, що створює звукові ефекти.

Кожен підрозділ цієї частини містить:

- Детальний опис кожного компонента, що використовується при реалізації пристрою. Це включає і пояснення їх вибору з технічної точки зору.
- Візуальні схеми та докладні інструкції щодо підключення електронних компонентів до мікроконтролера. Це включає як логічні розташування елементів, так і фізичні з’єднання на платі.
- Опис процесу створення кожної симуляції, результати тестування та аналіз отриманих даних.

### 3.2 Імітація спалаху блискавки

Для того, щоб симулювати блискавку, було використано світлодіод – напівпровідниковий пристрій, що при пропусканні через нього електричного струму випромінює світло. Для запобігання пошкодження як світлодіода, так і мікроконтролера, також використано резистор – електронний компонент, призначений для обмеження сили струму та розподілу напруги.

Перелік необхідних елементів:

- Мікроконтролер Arduino.
- Макетна плата.
- Світлодіод.
- Один резистор напругою в 220 Ом.
- З'єднувальні дроти «тато-тато».

Процес підключення:

По своїй структурі світлодіод в цілому схожий з випрямляючим діодом – у нього є 2 виводи: анод та катод (рис. 3.1). Ця особливість зумовлює необхідність дотримання полярності при його підключенні до джерела електричного струму. На практиці ж все просто – як правило, у ніжок світлодіода довжина є різною, довга ніжка є анодом, а коротка – катодом.

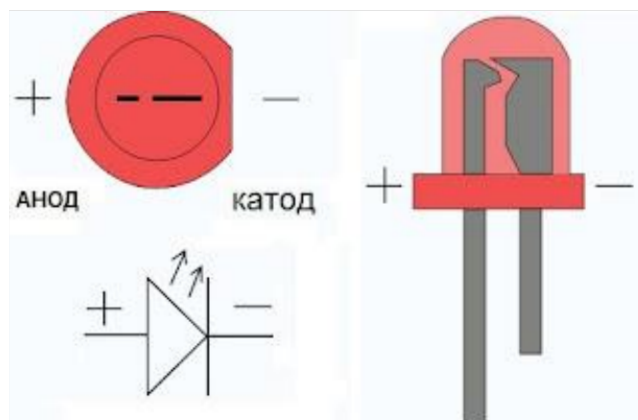


Рисунок 3.1 – Схема полярності світлодіода [28]

Анод світлодіода підключається через струмообмежувальний резистор до живлення, а катод – до «землі». Для розрахунку необхідного резистора потрібно

знати падіння напруги на світлодіоді (таблиця 3.1), його струм та напругу живлення.

Таблиця 3.1 – Падіння напруги для більшості індикаторних світлодіодів:

| Колір         | Напруга    |
|---------------|------------|
| Інфрачервоний | ~ 1.6В     |
| Червоний      | 1.6 – 2.0В |
| Помаранчевий  | 2.0 – 2.1В |
| Жовтий        | 2.1 – 2.2В |
| Зелений       | 1.9 – 3.5В |
| Синій         | 2.5 – 3.6В |
| Білий         | 2.8 – 4.0В |

Розрахунок опору резистора для світлодіода відбувався за формулою (3.1) [29]:

$$R = \frac{(U_s - U_f)}{I_f}, \quad (3.1)$$

де  $R$  – шуканий опір;

$U_s$  – напруга живлення, до якого було підключено світлодіод, в нашому випадку – 5В;

$U_f$  – падіння напруги на світлодіоді, в нашому випадку – 2.5 – 3.6В;

$I_f$  – сила струму світлодіода, в нашому випадку – 20мА.

Розраховуємо  $R = \frac{5-3}{0,02} = 100 \text{ Ом}$

Отже, для нашого прикладу підійде будь-який резистор опором від 100 Ом до 1 кОм.

Схема підключення наведена нижче у рисунку 3.2:

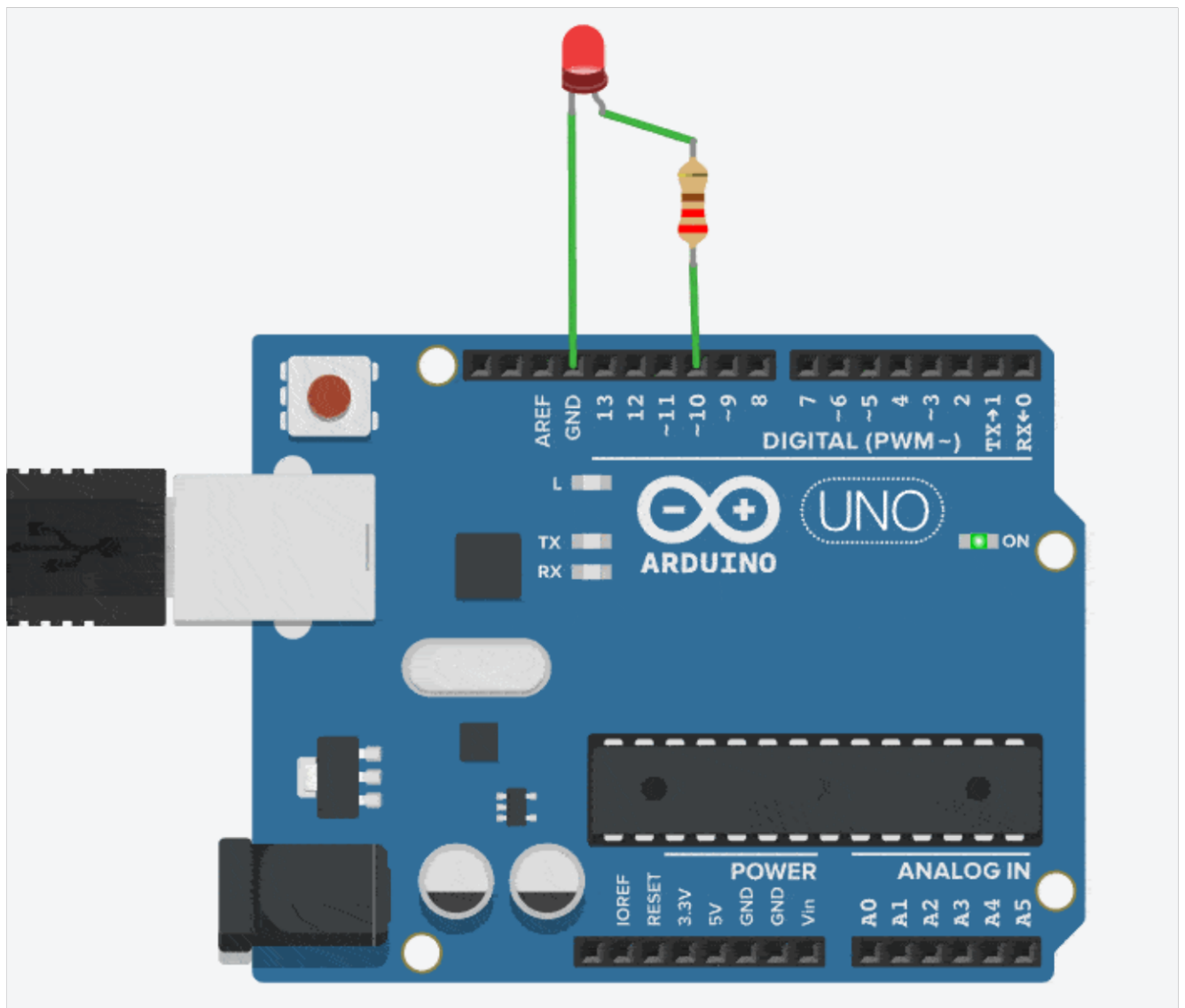


Рисунок 3.2 – Схема підключення світлодіода [29]

Програмний код (додаток А) контролює частоту та тривалість спалахів.

Результати виконання роботи:

Після підключення всіх компонентів до плати та завантаження коду спостерігаємо швидкі заблимування світлодіода світло-синього кольору тривалістю в 0.05 секунд кожне та засвітлення тривалістю в 1 секунду наприкінці кожного циклу, що імітує спалахи блискавки (рис. 3.3).

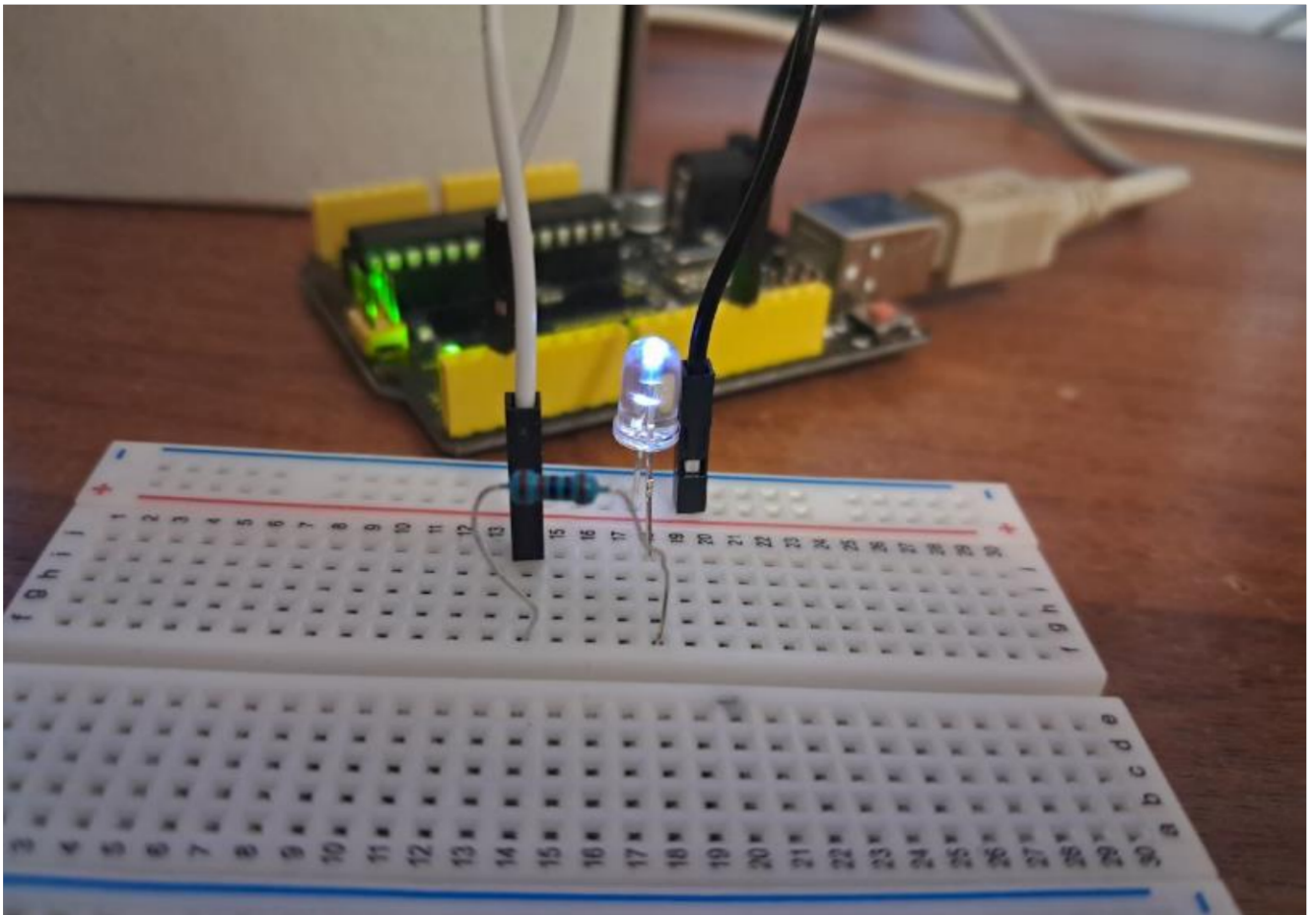


Рисунок 3.3 – Імітаційна модель спалаху блискавки

### 3.3 Симуляція сонячного світла

В даному випадку, симуляція сонячного світла має виконувати одразу 2 задачі: бути джерелом дуже яскравого світла і в той же час збільшувати свою температуру. Досягнути цього ефекту було вирішено шляхом використання в якості джерела світла лампу розжарювання напругою в 220В за наступних причин: лампи розжарювання виробляють світло, яке за своїм спектром та інтенсивністю наближається до сонячного, що забезпечує реалістичну симуляцію. Також при цьому лампи генерують разом зі світлом ще й тепло, що додатково допомагає імітувати сонячний вплив.

Перелік необхідних компонентів:

- Мікроконтролер Arduino.
- Макетна плата.
- Лампа розжарювання напругою 220В.

- Патрон для лампи.
- Реле (рис. 3.4).
- З'єднувальні дроти типу «тато-тато».
- Кнопка.

Процес підключення:

Реле – це електромеханічний пристрій, який використовує електромагніт для механічного перемикачання контактів. Це дозволяє керувати високострумними пристроями, використовуючи при цьому низьковольтні сигнали [18]. Його використання при розробці проекту обгрунтоване тим, що мікроконтролери, такі як Arduino, не можуть безпосередньо керувати високовольтними пристроями через обмеження в допустимих напрузі та струмі, а реле забезпечує безпечне розділення між низьковольтними сигналами управління та високовольтними ланцюгами.



Рисунок 3.4 – Реле, обране для розробки проекту [23]

Модуль дозволяє комутувати навантаження зі струмом до 30А. При цьому для керування реле достатньо 5мА струму керуючого сигналу. Для захисту входу і обмеження струму через оптрон гальванічної розв'язки на платі реле встановлений струмообмежувальний резистор.

З одного боку реле має 3 виходи – COM, NO, NC. Саме до цієї сторони і було підключено лампу розжарювання. COM (Common) підключення реле – це його рухома частина. Коли реле вимкнене, COM підключений до NC (Normally Closed). NO (Normally Open) підключення залишається вимкненим доки реле не буде увімкненим (рис. 3.5). Щоб лампа вмикалась тоді, коли вмикається реле – необхідно спочатку вкрутити її в патрон, один кінець якого підключається до NO, а інший кінець – є частиною проводу, який підключено до джерела живлення 220В. Другу ж частину проводу підключено до COM входу реле.

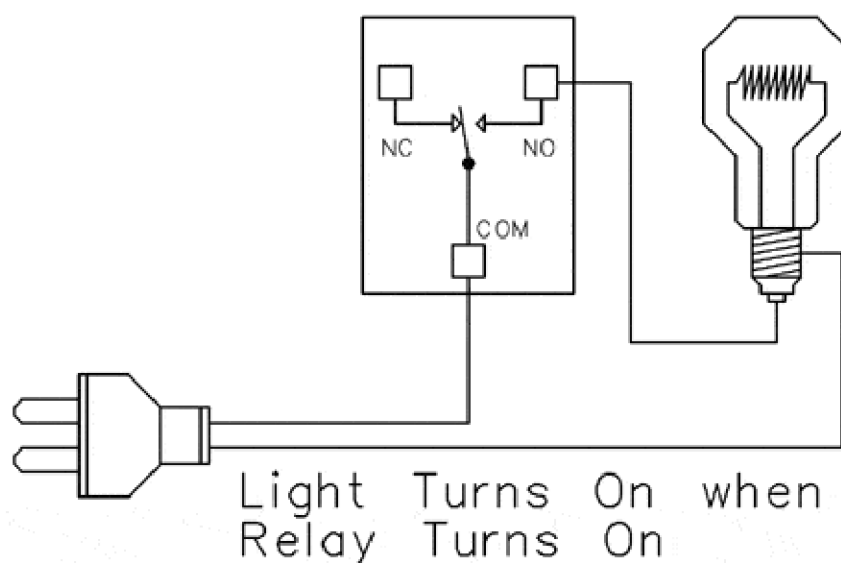


Рисунок 3.5 – Схема підключення лампи розжарювання до реле [30]

З іншого боку реле знаходяться входи DC+, DC-, VREF та CH1:

- DC+ є входом позитивного виводу джерела живлення;
- DC- є входом негативного виводу джерела живлення;
- VREF керує полярністю керуючого сигналу;
- CH1 є входом керуючого сигналу.

Для керування модулем сигналом позитивної полярності вхід VREF необхідно підключити до мінуса джерела живлення, а сигналом негативної полярності – до плюса. При підключенні реле до мікроконтролера UNO R3 (рис.3.6) було зроблено наступне:

- Під'єднано вхід CH1 реле до цифрового піна на мікроконтролері (в даному випадку – 13).
- З'єднано конекторами входи VREF та DC- реле та GND на мікроконтролері.
- Під'єднано вхід DC+ реле до джерела живлення (в даному випадку – пін 5V).
- Керування реле відбувалось за допомогою кнопки – її було підключено до цифрового піна (в даному випадку, 8) з одного боку, а з іншого – до «землі».

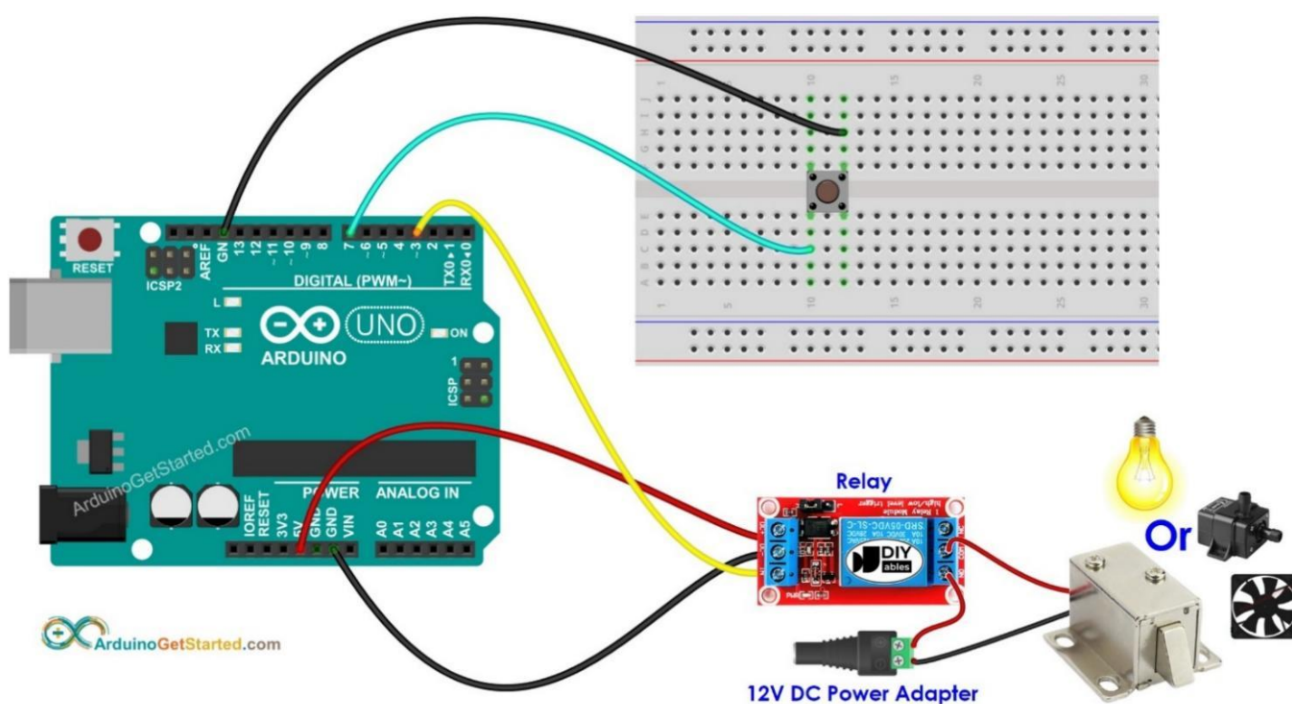


Рисунок 3.6 – Схема підключення реле до мікроконтролера [31]

Код програми (додаток Б) за допомогою кнопки керує реле: коли пін реле встановлюється в низький стан (LOW), реле активується і замикає контакт, дозволяючи електричному струму пройти до лампи, а у високому стані (HIGH) – деактивується та розмикає контакт, перериваючи струм.

Результати виконання роботи:

Після завантаження програмного коду в мікроконтролер, пристрій при натисканні кнопки подає високовольтні сигнали на лампу розжарювання, запалюючи її та таким чином імітуючи ефект сонячного світла (рис. 3.7).

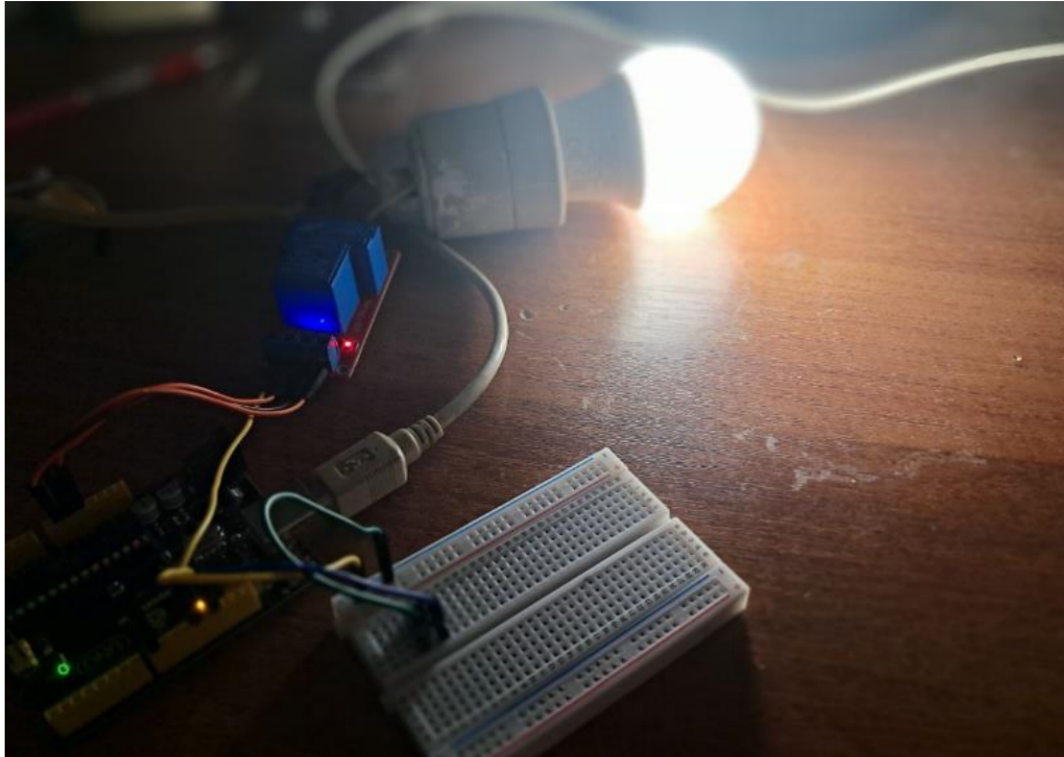


Рисунок 3.7 – Імітаційна модель сонячного світла

### 3.4 Симуляція ефекту землетрусу

Перелік необхідних компонентів:

- Вібромоторчик напругою 4.5В.
- Транзистор BC557.
- Резистор 10 Ом.
- Резистор 56 кОм.
- Мікроконтролер Arduino.
- Макетна плата.
- З'єднувальні дроти «тато-тато».

Було вирішено вибрати саме 4.5В вібромоторчик тому, що він є найбільш прямим і простим способом, адже він здатен створювати вібрації, які ефективно імітують ефект землетрусу. Також він є невеликим за розміром, що дозволяє інтегрувати його майже в будь-який проект. Низька напруга також є перевагою, адже це робить вібромоторчик безпечним та зручним у використанні.

Резистор 10 Ом обмежує струм, що проходить через транзистор та вібромоторчик, захищаючи компоненти від перенавантаження.

Резистор 56 кОм обмежує струм, що протікає через базу транзистора, забезпечуючи його правильне увімкнення та вимкнення.

Підключення компонентів:

Обраний вібромоторчик мав одну ваду – його проводи були занадто тонкими та м'якими, що унеможливило процес його підключення до макетної плати, тому було вирішено зробити наступне:

- За допомогою кухонної запальнички було оголено частину проводів вібромоторчика.

- Використовуючи канцелярський ніж було зачищено проводи від залишків ізоляції.

- За допомогою пінцету кожен з проводів був щільно скручений з ніжками, відрізаними у резистора.

Після виконання всіх вищезазначених дій база стала досить твердою, щоб її можна було з'єднати з макетною платою. Щоб заживити вібромоторчик від мікроконтролера, необхідно один з його проводів підключити до джерела живлення (в даному випадку – 5В), а інший – до колектора транзистора.

Транзистор – це напівпровідниковий тріод, здатний при невеликому вхідному сигналі керувати значним струмом вихідного ланцюгу, що дозволяє використовувати його для підсилення, генерування, комутації чи перетворення електричного сигналу. Так як вібромоторчик потребує більше струму, ніж може забезпечити вихідний пін мікроконтролера, необхідно проводити струм через транзистор. Використання саме BC557 зумовлено тим, що він є PNP біполярним транзистором, а отже, він проводить струм, коли база знаходиться на нижчому потенціалі, ніж емітер. У даному випадку, коли мікроконтролер подає низький сигнал на базу транзистора, він дозволяє струму протікати з колектора до емітера, що вмикає вібромоторчик (рис. 3.8).

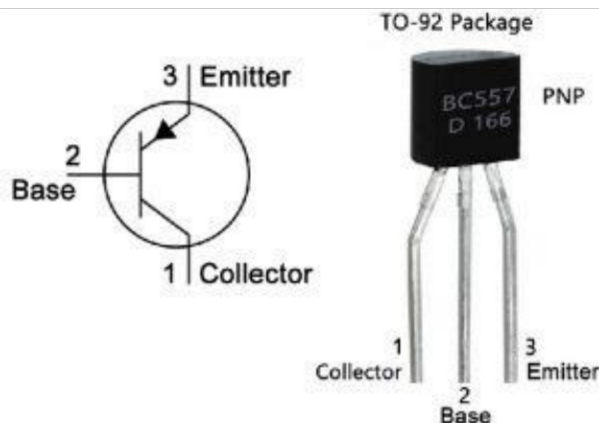


Рисунок 3.8 – Розпіновка транзистора BC557 [33]

При його підключенні до плати, було зроблено наступне:

- Колектор транзистора через резистор 10Ом підключено до будь-якого з контактів вібромоторчику.
- Емітер з'єднано з «землею» (GND) мікроконтролера.
- Базу транзистора через 56кОм резистор підключено до цифрового піна (в даному випадку – 7).

Візуально схему підключення пристрою зображено в рисунку 3.9:

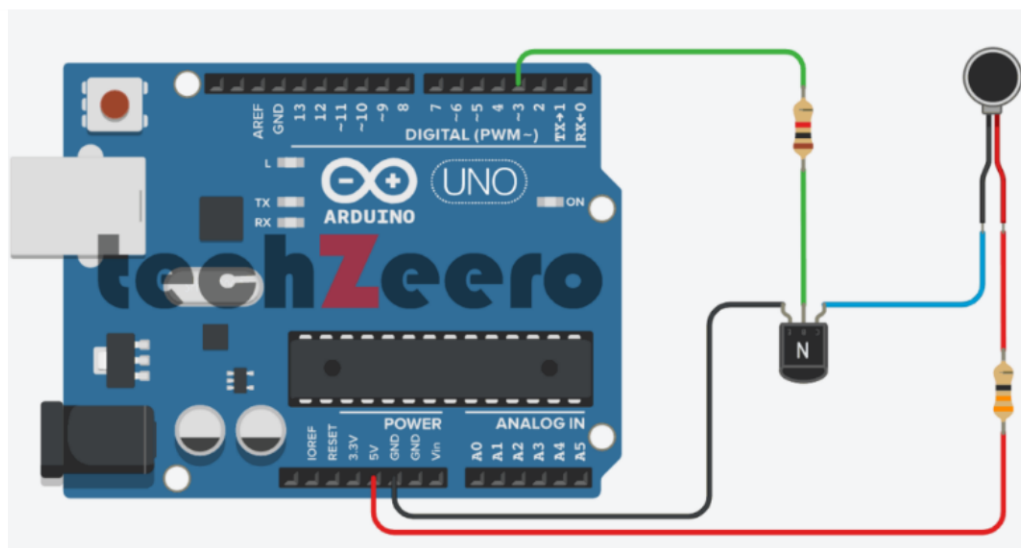


Рисунок 3.9 – Схема підключення вібромоторчика до контролера [32]

Код програми (додаток В) керує цифровим піном 7 – у високому (HIGH) стані транзистор BC557 відкривається і дозволяє струму протікати через вібромоторчик,

викликаючи його вібрацію, а в низькому (LOW) стані – транзистор закривається, перериваючи струм та зупиняючи вібрацію.

Результати виконання роботи:

Після завантаження програмного коду в мікроконтролер, циклічно, з інтервалами в одну секунду, вібромоторчик вмикається й вимикається, емулюючи ефект землетрусу (рис. 3.10). Використання резисторів та транзисторів дозволяє забезпечити стабільну роботу пристрою.

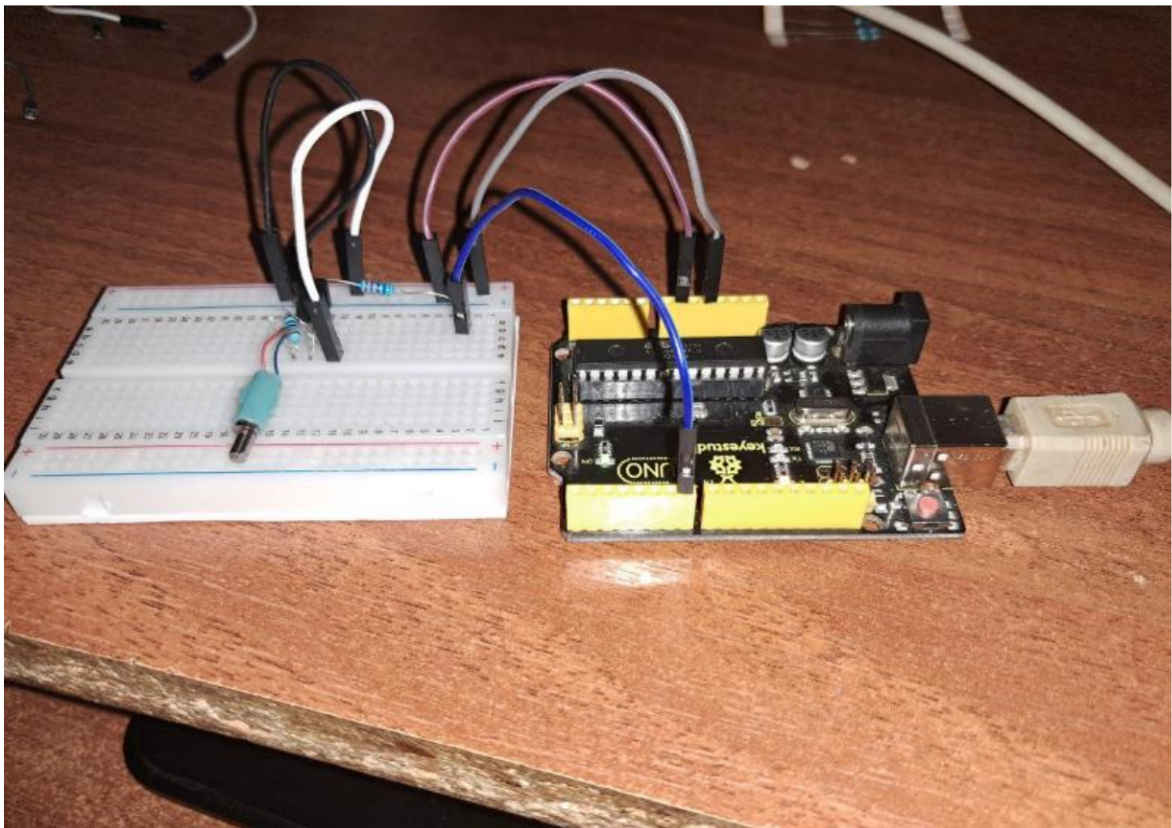


Рисунок 3.10 – Імітаційна модель ефекту землетрусу

### 3.5 Симуляція ефекту вітру

Перелік компонентів:

- Біполярний транзистор.
- Резистор 220 Ом.
- 12В кулер.
- Стороннє 12В джерело живлення.

- Мікроконтролер Arduino.
- Макетна плата.
- З'єднувальні дроти «тато-тато».

Було вирішено обрати саме мотор тому, що він здатен забезпечувати значну потужність, необхідну для обертання кулера, що створює відчутний потік повітря. Також важливою перевагою є те, що мотор дозволяє регулювати швидкість обертання, що дає можливість контролювати силу створюваного вітру.

Обертання кулера є найбільш простим та ефективним способом створити фізичний ефект вітру, адже він є досить недорогим та при цьому доступний в різних розмірах та конфігураціях.

Біполярні транзистори, такі, як NPN або PNP дозволяють керувати великими струмами за допомогою малих керуючих сигналів від мікроконтролера. Використати можна той ж самий BC557, через який відбувалось заживлення вібромоторчика.

Резистор 56 кОм виконує таку ж функцію, як у проекті імітації землетрусу – обмеження струму, що протікає через базу транзистора для забезпечення правильної його роботи.

Було використано стороннє джерело живлення адже плата UNO R3 не може генерувати 12В струм, необхідний для заживлення кулера. Також його використання зробило пристрій автономним – йому не потрібно підключення до комп'ютера через USB для отримання енергії. В якості джерела струму було використано батареї бренду Duracell.

Підключення компонентів:

Один контакт мотора підключено до колектора транзистора, а інший – до позитивного контакту джерела живлення. До осі мотора механічно прикріплено кулер.

Колектор біполярного транзистора підключено до одного з контактів мотора, емітер підключено до «землі» (GND), а базу – до резистора 56 кОм.

Резистор одним кінцем підключено до бази транзистора, а іншим – до цифрового піна на мікроконтролері Arduino (в даному випадку – 4).

Стороннім джерелом живлення подано струм на лінії позитивного та негативного струму на макетній платі згідно полярностей (рис. 3.11).

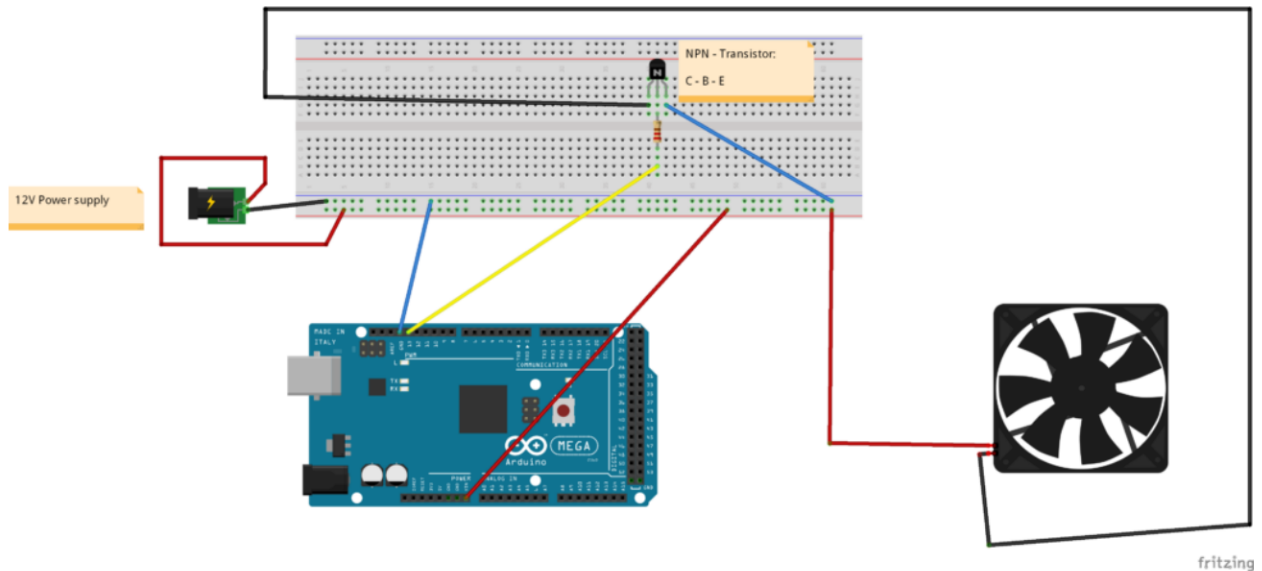


Рисунок 3.11 – Схема підключення кулера [34]

Код програми (додаток Г) керує пінами 4 та 9. Коли пін 4 встановлений у високий стан (HIGH), транзистор відкривається, дозволяючи струму проходити через мотор, що викликає його обертання. Швидкість мотора регулюється за допомогою ШІМ сигналу на пині 9. В даному випадку швидкість мотора було встановлено на 250 з можливих 255.

Після завантаження програмного коду в мікроконтролер, пристрій з можливістю налаштування швидкості та напрямку руху виконував оберти кулера, що створював фізичний потік повітря, симулюючи ефект вітру.

### 3.6 Симуляція звуку грому

Перелік компонентів:

- Зумер на 5В.
- Резистор 100Ом.
- Мікроконтролер.

- Макетна плата.
- З'єднувальні дроти «тато-тато».

Підключення компонентів:

Зумер – найпростіший модуль для отримання звуку частотою близько 2 кГц. Дуже часто використовується в електроніці для звукової індикації: будильники, таймери, мікрохвильовки, пральні машини та інші предмети побуту. Також дуже часто використовується на материнських платах для персональних комп'ютерів – там він за допомогою звукових сигналів повідомляє користувача про тип помилок при POST-фазі запуску. Поділяються зумери на 2 типи – активні та пасивні. Активні зумери при подачі на них струму одразу починають видавати звук, адже мають вбудований генератор. Пасивний ж зумер просто клацне та все, адже за відсутності генератора звук він може видавати лише при постійній подачі струму, зате тональність звуку можна налаштовувати програмно.

Зумер потребує від 10 до 30 мА, тому в принципі його можна підключати до цифрового піна напругу, але якщо робота йде з мікроконтролерами по типу ESP8266, варто використовувати струмообмежуючий резистор з опором в 100Ом. Для правильного підключення зумера, як і в випадку зі світлодіодом, потрібно дотримуватись полярності (рис. 3.12).



Рисунок 3.12 – Полярність звукового модуля [35]

Довгу ніжку (анод) потрібно підключити до цифрового піна на платі мікроконтролера (в даному випадку – 3), а коротку (катод) – до «землі» (рис. 3.13).

За необхідності треба подавати струм через резистор, що встановлюється між анодом та цифровим піном.

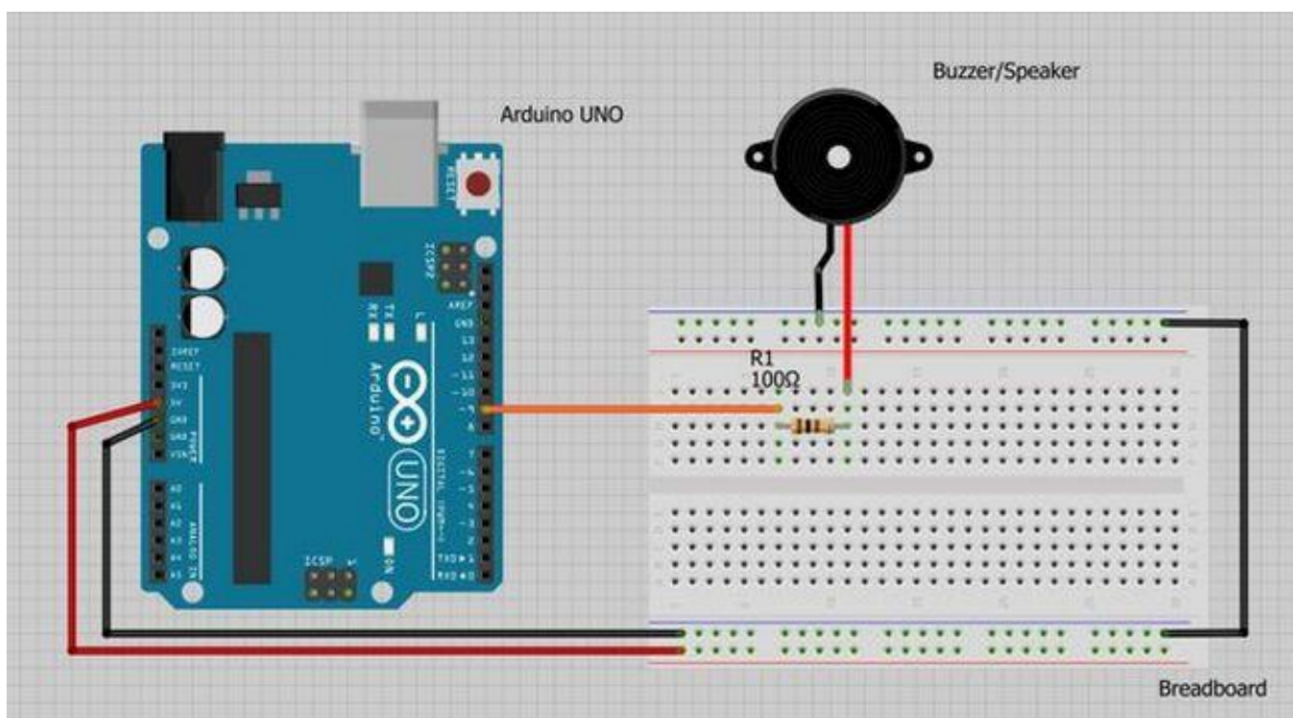


Рисунок 3.13 – Схема підключення зумера до мікроконтролера [36]

Програмний код (додаток Д) задає цифровий пін 3 як вихід. Саме його було використано для виводу сигналу на зумер. Коли на пін 3 подається високий (HIGH) сигнал, зумер отримує напругу та починає створювати звук, а коли низький – звук зупиняється.

Результат виконання роботи:

При завантаженні програмного коду на мікроконтролер, зумер створює звук за патерном, що починається з різкого удару, а потім поступово згасає з нерівномірними коливаннями, що характерно для справжнього грому (рис. 3.14)

- Перша серія з трьох коротких, різких ударів, які імітують початковий удар. Кожен з ударів триває 50 мілісекунд з паузами по 100 мілісекунд між ударами.
- Коротка пауза, тривалістю в 300 мілісекунд.
- Друга серія з 5 ударів, у яких менша інтенсивність та довші паузи – кожен триває 100 мілісекунд, а паузи збільшені до 200.

- Пауза в 600 мілісекунд, що імітує відлуння.
- Третя серія з трьох ударів, що затухають, та з ще довшими паузами.

Кожен удар триває 200 мілісекунд, а паузи – 300.

- Після всіх серій додається пауза в 2 секунди щоб імітувати інтервал між різними ударами грому.

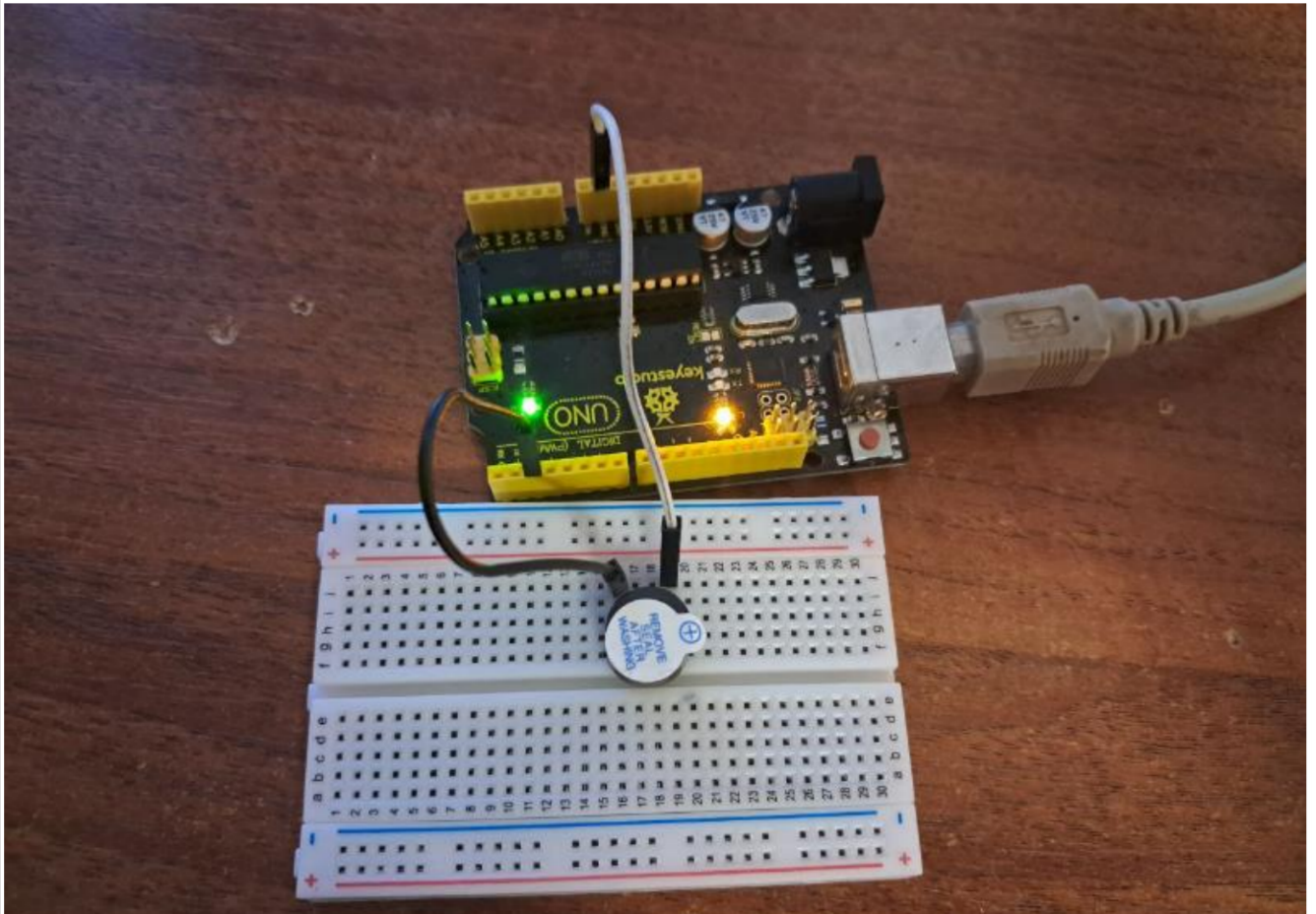


Рисунок 3.14 – Імітаційна модель звуку грому

## ВИСНОВКИ

В ході дослідження було реалізовано пристрій, здатний симулювати погодні умови, такі, як: спалах блискавки, удар грому, сонячні промені, землетрус та пориви вітру.

Проведено аналіз терміну та класифіковано види симуляцій погодних умов: програмні та апаратні. Визначено області застосування пристроїв та сформовано історичні чинники, що сприяли їх розвитку в минулому і сприятимуть у майбутньому. Оглянуто популярні платформи та інструменти для симуляцій погодних умов.

Визначено фактори, згідно з якими обиралось апаратне забезпечення для реалізації проекту. Проаналізовано технічні характеристики мікроконтролера та досліджено його структуру на прикладах системи живлення, периферійних пристроїв, основних та альтернативних функцій кожного з цифрових виходів. Оглянуто найбільш популярні середовища програмування, виокремлено їх основні переваги та недоліки.

Розроблено програмне забезпечення для керування симуляціями різних погодних умов з використанням наступних електронних компонентів: резистори, транзистори, реле, зумер, вібромоторчик, світлодіод, кулер.

Описано процес підключення компонентів, наведено ілюстративний матеріал. Пристрій було протестовано й налагоджено, а результати його роботи – задокументовано.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. PhET Інтерактивне моделювання. URL: <https://phet.colorado.edu/uk/> (дата звернення: 10.02.2024).
2. С. М. Андрєєв, С. І. Горелик, А. С. Нечаусов, Д. К. Саул-Гозе Застосування геоінформаційних технологій для побудови картографічних моделей небезпечних метеорологічних явищ. Харків, 2022. 12 с. URL: <https://doi.org/doi:10.26906/SUNZ.2022.1.004> (дата звернення: 10.02.2024).
3. Закон Бернуллі. URL: [https://uk.wikipedia.org/wiki/Закон\\_Бернуллі](https://uk.wikipedia.org/wiki/Закон_Бернуллі) (дата звернення: 10.02.2024).
4. Історія науки про зміни клімату. URL: [https://uk.wikipedia.org/wiki/Історія\\_науки\\_про\\_зміни\\_клімату](https://uk.wikipedia.org/wiki/Історія_науки_про_зміни_клімату) (дата звернення: 10.02.2024).
5. Жан Батист Жозеф Фур'є. URL: [https://uk.wikipedia.org/wiki/Жан\\_Батист\\_Жозеф\\_Фур'є](https://uk.wikipedia.org/wiki/Жан_Батист_Жозеф_Фур'є) (дата звернення: 10.02.2024).
6. Математичне моделювання або як метеорологи досліджують атмосферу. URL: <https://expedicia.org/matematichne-modelyuvannya-abo-yak-meteo/> (дата звернення: 10.02.2024).
7. Advances in Real-Time Rendering in 3D Graphics and Games. URL: <https://www.advances.realtimerendering.com> (дата звернення: 10.02.2024).
8. Simulation-based education involving online and on-campus models in different European universities - International Journal of Educational Technology in Higher Education. URL: <https://educationaltechnologyjournal.springeropen.com/articles/10.1186/s41239-020-0181-у> (дата звернення: 10.02.2024).
9. Climate Models. URL: <https://climate.mit.edu/explainers/climate-models> (дата звернення: 10.02.2024).
10. Як виробляють штучний сніг: процес і технології. URL: <https://ukr.zapisi.cx.ua/yak-viroblyayut-shtuchniy-snig-proces-i-tekhnologii/> (дата звернення: 10.02.2024).

11. Weather Research & Forecasting Model (WRF) | Mesoscale & Microscale Meteorology. URL: <https://www.mmm.ucar.edu/models/wrf> (дата звернення: 10.02.2024).
12. Global Forecast System (GFS). URL: <https://www.ncei.noaa.gov/products/weather-climate-models/global-forecast> (дата звернення: 10.02.2024).
13. Environmental Modeling Center (EMC). URL: <https://www.emc.ncep.noaa.gov/emc.php> (дата звернення: 10.02.2024).
14. Погодні мапи. Вид із супутника і метеорологічний радар наживо. URL: <https://www.meteoblue.com/uk/weather/maps> (дата звернення: 10.02.2024).
15. Arduino Uno Rev3 від Keyestudio. URL: <https://arduino.ua/prod2493-arduino-uno-rev3-ot-keyestudio> (дата звернення: 10.02.2024).
16. Microchip Studio for AVR and SAM Devices. URL: <https://www.microchip.com/en-us/tools-resources/develop/microchip-studio> (дата звернення 10.02.2024)
17. VisualMicro – Arduino IDE For Visual Studio. URL: <https://www.visualmicro.com/> (дата звернення: 12.02.2024).
18. Реле URL: <https://uk.wikipedia.org/wiki/Реле> (дата звернення: 12.02.2024).
19. Симуляція погодних умов: ентузіаст створив чудернацький механізм у Minecraft – відео. URL: [https://games.24tv.ua/simulyatsiya-pogodnih-umov-minecraft-chudernatskiy-novini-sogodni\\_n1572429](https://games.24tv.ua/simulyatsiya-pogodnih-umov-minecraft-chudernatskiy-novini-sogodni_n1572429) (дата звернення: 12.02.2024).
20. Enviro - Sky and Weather. Unity Asset Store. URL: <https://assetstore.unity.com/packages/tools/particles-effects/enviro-sky-and-weather-33963> (дата звернення: 12.02.2024).
21. Двоканальний димер 220В для Arduino від RobotDyn URL: <https://arduino.ua/prod3118-dimmer-220v-2-kanala-dlya-arduino-ot-robotdyn> (дата звернення: 12.02.2024).
22. Інтернет речей (IoT) в сільському господарстві: 9 прикладів використання технологій для точного землеробства (і виклики, які слід врахувати). URL: <https://www.agrilab.ua/internet-rechej-iot-v-silskomu-gospodarstvi-9-prykladiv->

vykorystannya-tehnologij-dlya-tochnogo-zemlerobstva-i-vyklyky-yaki-slid-vrahuvaty/  
(дата звернення: 12.02.2024).

23. Модуль потужного реле 5В 30А з опторозв'язкою.  
URL: <https://arduino.ua/prod1767-1-channel-5v-30a-high-power-relay-module-for-arduino> (дата звернення: 12.02.2024).

24. Arduino UNO - популярна плата розробки  
URL: <https://itmaster.biz.ua/directory/kits-nabory/arduino-uno.html> (дата звернення: 13.02.2024).

25. Arduino UNO: A Short Review URL: <https://www.edugonist.com/arduino-uno/> (дата звернення: 13.02.2024).

26. Урок 2. Плата Arduino UNO R3. Опис, характеристики.  
URL: <http://www.radiocomponents.com.ua/index.php/articles/12-arduino/arduino-lessons/16-urok-2-plata-arduino-uno-r3-opisanie-kharakteristiki> (дата звернення: 13.02.2024).

27. IT Master – Інформаційний ресурс про електроніку та програмування.  
URL: [https://itmaster.biz.ua/ftp/schematics/arduino\\_uno/arduino\\_uno\\_r3\\_schematic.pdf](https://itmaster.biz.ua/ftp/schematics/arduino_uno/arduino_uno_r3_schematic.pdf)  
(дата звернення: 13.02.2024).

28. Світлодіод. URL: [https://radiokonstruktor.blogspot.com/2017/10/blog-post\\_22.html](https://radiokonstruktor.blogspot.com/2017/10/blog-post_22.html) (дата звернення: 14.02.2024).

29. Arduino Blink блимаєм світлодіодом URL: <https://qazf.com.ua/arduino-blink-platformio/> (дата звернення: 14.02.2024).

30. Relay Logic URL: <https://ncd.io/blog/relay-logic/> (дата звернення: 14.02.2024).

31. Arduino - Button - Relay  
URL: <https://arduinogetstarted.com/tutorials/arduino-button-relay> (дата звернення: 14.02.2024).

32. Vibration Motor with Arduino - Code, Circuit for Vibration Motor.  
URL: <https://techzeero.com/arduino-tutorials/vibration-motor-with-arduino/> (дата звернення: 14.02.2024).

33. BC557 Transistor : Pin Configuration, Specifications & Its Applications.  
URL: <https://www.elprocus.com/bc557-transistor/> (дата звернення: 15.02.2024).

34. How to build an Arduino fan controller. Step by Step explanation.  
URL: <https://nerd-corner.com/arduino-fan-controller/> (дата звернення: 15.02.2024).

35. Buzzer : Working, Types, Circuit, Advantages & Disadvantages.  
URL: <https://www.elprocus.com/buzzer-working-applications/> (дата звернення: 15.02.2024).

36. How to use a buzzer with Arduino.  
URL: <https://www.ardumotive.com/how-to-use-a-buzzer-en.html> (дата звернення: 15.02.2024).

# ДОДАТКИ

## Додаток А

### Код для керування світлодіодом

```
void setup () {  
  pinMode(13, OUTPUT);  
  
}  
  
void loop() {  
  
  digitalWrite(2, HIGH);  
  delay(50);  
  digitalWrite(2, LOW);  
  delay(50);  
  digitalWrite(2, HIGH);  
  delay(50);  
  digitalWrite(2, LOW);  
  delay(50);  
  digitalWrite(2, HIGH);  
  delay(50);  
  digitalWrite(2, LOW);  
  delay(50);  
  digitalWrite(2, HIGH);  
  delay(50);  
  digitalWrite(2, LOW);  
  delay(1000);  
}
```

## Додаток Б

### Код для керування лампою розжарювання

```
int buttonPin = 8;
int relayPin = 12;
int val = 0;
int lightON = 0;
int pushed = 0;

void setup() {
  Serial.begin(9600);
  pinMode(buttonPin, INPUT_PULLUP);
  pinMode(relayPin, OUTPUT);
  digitalWrite(relayPin, HIGH);
}

void loop() {
  val = digitalRead(buttonPin);

  if(val == HIGH && lightON == LOW) {
    pushed = 1 - pushed;
    delay(100);
  }

  lightON = val;

  if(pushed == HIGH) {
    Serial.println("Light ON");
    digitalWrite(relayPin, LOW);
  } else {
    Serial.println("Light OFF");
    digitalWrite(relayPin, HIGH);
  }

  delay(100);
}
```

## Додаток В

### Код для керування вібро моторчиком

```
void setup() {  
  pinMode(7, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(7, HIGH);  
  delay(1000);  
  digitalWrite(7, LOW);  
  delay(1000);  
}
```

## Додаток Г

### Код для симуляції вітру

```
int rotationSpeed = 9;
int state = 4;

void setup(void)
{
  pinMode(state,OUTPUT);
  pinMode(rotationSpeed,OUTPUT);
}
void advance(void)
{
  digitalWrite(state,HIGH);
  analogWrite(rotationSpeed,250);
}
void loop()
{
  advance();
}
```

## Додаток Д

### Код для симуляції ударів грому

```
void setup() {
  pinMode(3, OUTPUT);
}

void loop() {
  for (int i = 0; i < 3; i++) {
    digitalWrite(3, 1);
    delay(50);
    digitalWrite(3, 0);
    delay(100);
  }

  delay(300);

  for (int i = 0; i < 5; i++) {
    digitalWrite(3, 1);
    delay(100);
    digitalWrite(3, 0);
    delay(200);
  }

  delay(600);

  for (int i = 0; i < 3; i++) {
    digitalWrite(3, 1);
    delay(200);
    digitalWrite(3, 0);
    delay(300);
  }

  delay(2000);
}
```