ЗМІСТ

1. Аналітичний огляд мікроконтролера фірми AI-Thinker 9

1.1 Мікроконтролер 9

1.2 Компанiя AI-Thinker 10

1.3 Загальні відомості про RISC і CISC МК 11

1.5 Засоби розробки 16

2. Аналітичний огляд мікроконтролерної плати Node MCU 17

2.1 NodeMCU 17

2.2 Розміри і габарити плати 18

2.3 Живлення 18

2.4 Входи і виходи 18

3. Огляд вхідних та вихідних ланцюгів 22

3.1 Огляд P10 RGB Matrix дiсплея. 22

3.3 Технічні характеристики дисплея 23

3.4 Датчик DHT11 23

3.5 Технічні характеристики DHT11 24

4. Розробка структурної схеми мікропроцесорної системи 25

5. Розробка принципової схеми мікропроцесорної системи 26

5.1 Принципова електрична схема 26

5.2 Fritzing  26

6. Розробка алгоритму роботи мікропроцесорної системи 29

7. Розробка програмного забезпечення для мікропроцесорної системи 30

Висновок 33

Перелік використаних джерел 34

**ВСТУП**

У наш час програмування є однією з найбільш динамічно розвиваючихся галузей, що відображає світові тенденції технологічного прогресу та потреби суспільства. Однією з найактуальніших галузей програмування є аналіз даних з соціальних медіа. Соціальні платформи, такі як Facebook, Instagram, Twitter та інші, стали не просто майданчиками для спілкування, але й важливим інструментом для бізнесу, політики, науки та культури. Вони об'єднують мільйони користувачів з усього світу, дозволяючи їм обмінюватися інформацією, думками та переживаннями.

Актуальність розробки програмного забезпечення для аналізу даних з соціальних медіа визначається не лише через загальний розвиток цих платформ, але й через необхідність ефективного управління контентом та взаємодії з аудиторією. Компанії, маркетологи, політики та дослідники активно використовують дані з соціальних медіа для аналізу тенденцій, прогнозування поведінки аудиторії, а також для створення та вдосконалення своїх стратегій.

Метою нашої роботи є розробка програми для аналізу даних з соціальних медіа, спрямованої на вивчення популярності та взаємодії з контентом. Важливими завданнями програми є виявлення ключових трендів, аналіз залученості аудиторії та визначення ефективності контенту.

Галузь застосування програми включає маркетингові дослідження, наукові аналізи, стратегії управління контентом та багато іншого. Ми впевнені, що наша розробка стане корисним інструментом для студентів, дослідників та фахівців у сфері соціальних медіа, сприяючи їхньому подальшому професійному розвитку та науковим досягненням.

Програмне забезпечення було створене в Visual Studio Code, інструменті для розробки, редагування та налагодження сучасних веб-застосунків і програм для хмарних систем. Visual Studio Code доступний безкоштовно для платформ Windows, Linux і OS X. Microsoft презентувала Visual Studio Code у квітні 2015 на конференції Build 2015 як перший кросплатформний продукт у лінійці Visual Studio.

Python - інтерпретована, об'єктно-орієнтована мова програмування з високорівневою синтаксичною структурою. Це мова загального призначення, яка найчастіше використовується для розробки веб-застосунків, наукових досліджень, штучного інтелекту та інших задач. Python має простий синтаксис, що дозволяє розробникам швидко писати чистий та зрозумілий код. Він підтримує модульність та розширюваність, що дозволяє легко інтегрувати з іншими мовами та бібліотеками.

**1. АНАЛІЗ ЗАДАЧІ, ЗАСОБІВ ТА МЕТОДІВ ЇЇ ВИРІШЕННЯ**

**1.1 Розумiння вимог**

**Вхідні дані:**

Програма для аналізу даних з соціальних медіа призначена для роботи зі структурованими даними у форматі CSV. Вона приймає дані через командний рядок де користувачем вказується шлях до відповідного файлу. Передбачається, що формат CSV-файлу має стандартизовану структуру, де зберігається інформація про публікації в соціальних мережах, таку як текст публікації, кількість лайків, коментарів та репости.

**Обробка даних:**

Після отримання вхідних даних у форматі CSV програма проводить їх обробку для подальшого аналізу. Цей процес включає зчитування даних з файлу та їх трансформацію у об’єкти класу. Після завершення обробки даних формується масив, що містить інформацію про кожну публікацію, включаючи текст, кількість лайків, коментарів та репости.

**Аналіз популярності:**

Одним із основних функціональних елементів програми є аналіз популярності контенту, що дозволяє визначити відгуки аудиторії на публікації. Програма використовує метрики, такі як загальна кількість взаємодій (лайки, коментарі, репости разом) або сортування постiв для оцінки популярності контенту. Такий аналіз дозволяє користувачеві краще розуміти, який контент найбільш цікавий для аудиторії та як взаємодіється з нею.

**1.2 Аналіз вихідних даних**

CSV-файл містить дані про різні публікації, що є повідомленнями або оголошеннями на платформі соціальних медіа чи іншому веб-сайті. Ось основнi стовпчики що буде мати CSV-файл:

1. **Назва публікації (Post title)**: Це перше поле в кожному рядку і містить заголовок або опис публікації.
2. **Лайки (Likes)**: Це кількість користувачів, які виразили своє задоволення публікацією, натиснувши кнопку "лайк".
3. **Коментарі (Comments)**: Це кількість коментарів, які були залишені під публікацією.
4. **Репости (Reposts)**: Це кількість разів, коли публікація була поділена іншими користувачами на їх власних сторінках чи профілях.
5. **Відвідування профілю (Profile visits)**: Це кількість разів, коли користувачі переходили на профіль автора публікації, можливо, після перегляду публікації.

Ці дані можуть бути використані для аналізу популярності публікацій, їхньої взаємодії з аудиторією та впливу на сторінку або профіль у соціальних мережах. Наприклад, високі цифри лайків та репостів можуть свідчити про успішні публікації, які викликали зацікавленість аудиторії. Коментарі також можуть дати уявлення про взаємодію з аудиторією та відгуки на контент.

**Загальнi вiдомостi про CSV:**

CSV (Comma-Separated Values) - це текстовий формат для зберігання даних у вигляді таблиці, де кожен рядок представляє один запис, а кожний стовпець в цьому записі містить значення різних полів, розділені комами або іншими розділовими знаками. CSV файл може мати розширення .csv.

Принцип роботи CSV файлу досить простий:

1. **Розділення даних**: Дані у CSV файлі зазвичай розділяються комами, хоча іноді використовуються інші розділові знаки, такі як крапки з комою або табуляція.
2. **Рядки і стовпці**: Кожен рядок у CSV файлі відповідає окремому запису або рядку даних, а кожен стовпець в рядку містить певну інформацію, яка відноситься до цього запису. Наприклад, якщо CSV файл містить дані про співробітників, кожен рядок може містити прізвище, ім'я, посаду тощо, а кожен стовпець у рядку буде відповідати конкретному полю (прізвище, ім'я, посада тощо).
3. **Кодування даних**: Дані у CSV файлах можуть бути закодовані в різних форматах, таких як UTF-8 або ASCII, залежно від потреби і використовуваних програм.
4. **Заголовок**: У деяких випадках перший рядок CSV файлу може містити заголовки стовпців, які надають інформацію про те, які дані містяться в кожному стовпці. Це полегшує читання та обробку даних.
5. **Зберігання даних**: CSV файли легко зберігаються у текстовому форматі і можуть бути відкриті та редаговані з використанням різноманітних програм, включаючи електронні таблиці, текстові редактори та спеціалізовані програми для обробки даних.

CSV файли широко використовуються для обміну даними між програмами, так як вони є простими у використанні та зрозумілими як для комп'ютерів, так і для людей.

**1.3 Вибір технологій**

Python - це високорівнева, інтерпретована мова програмування загального призначення з простим синтаксисом та динамічною типізацією.

#### **Загальна інформація:**

Python був розроблений Гвідо ван Россумом та вперше випущений у 1991 році. Він швидко став популярним завдяки своїй простоті та читабельності коду. Python підтримує багато парадигм програмування, таких як процедурне, об'єктно-орієнтоване, функціональне та аспектно-орієнтоване програмування.

#### **Основні особливості Python:**

1. **Простота читання та написання коду**: Python має чітку та лаконічну синтаксичну структуру, що дозволяє легко читати та розуміти код.
2. **Динамічна типізація**: Змінні в Python не пов'язані з конкретними типами даних під час оголошення, а типи визначаються автоматично під час виконання програми.
3. **Широкий спектр бібліотек та фреймворків**: Python має велику кількість стандартних бібліотек та сторонніх модулів, які полегшують розробку різноманітних проектів.
4. **Портативність**: Python працює на різних операційних системах, таких як Windows, macOS, Linux та інших.
5. **Швидкість розробки**: Благодаря простоті та потужним інструментам, розробка програм на Python зазвичай займає менше часу порівняно з іншими мовами програмування.
6. **Широке застосування**: Python використовується у багатьох галузях, включаючи веб-розробку, наукові обчислення, штучний інтелект, обробку даних, машинне навчання та багато інших.

### **Cторонні бібліотеки:**

У цьому проекті використовується лише модуль operator та функція attrgetter. Модуль operator є частиною стандартної бібліотеки Python і містить функції, які дозволяють виконувати операції з типовими операторами. Функція attrgetter з цього модулю використовується для отримання значень атрибутів об'єктів, що дозволяє здійснювати сортування за цими значеннями.

Це рішення обрано для того, щоб зберегти проект простим та не додавати надмірної складності за рахунок використання сторонніх бібліотек.

**Редактор коду:**

Як редактор коду, в цьому проектi був обраний Microsoft Visual Studio Code. **Visual Studio Code (VS Code)** - це безкоштовний редактор коду, який надає широкі можливості для розробки програмного забезпечення. Дане середовище розробки засноване на платформі Electron, що дозволяє використовувати вбудований веб-браузер і забезпечує можливості розширення за допомогою плагінів. VS Code підтримує різні мови програмування, включаючи Python, та надає багато функцій, які сприяють продуктивності розробника.

### Де використовують Visual Studio Code:

* Розробка веб-застосунків (Frontend та Backend).
* Розробка мобільних додатків.
* Розробка хмарних додатків.
* Розробка штучного інтелекту, машинного навчання та аналізу даних.
* Робота з системами керування версіями (наприклад, Git).
* Робота з Docker і Kubernetes.

### Плюси Visual Studio Code:

1. **Безкоштовність**: VS Code є безкоштовним для використання та має відкритий вихідний код.
2. **Підтримка різних мов програмування**: Включаючи Python, JavaScript, TypeScript, C++, Java, і багато інших.
3. **Розширюваність**: За допомогою розширень можна розширити функціональність редактора.
4. **Широкі можливості налаштування**: Велика кількість налаштувань та можливість налаштування редактора під власні потреби.
5. **Зручний інтерфейс користувача**: Інтуїтивно зрозумілий та легкий у використанні інтерфейс.

**2. ПРОЕКТУВАННЯ ЗАГАЛЬНОГО АЛГОРИТМУ РОБОТИ ПРОГРАМИ**

Для зручності рекомендується розбити роботу на два етапи.

**Перший етап** – створення коду, який буде переводити кожен рядок CSV-файлу в об'єкт класу Experiment для подальшої роботи.

У цьому етапі розроблюється клас FileManager, який матиме два методи:

* Метод імпортування, що приймає задане користувачем ім'я файлу та відкриває його стандартними засобами Python.
* Метод, який розділяє дані, що знаходяться в файлі, та одразу заносить їх до ініціалізатора об'єкта класу Post.

**Другий етап** – сортування масиву перетворених об'єктів та виведення їх в консоль.

На другому етапі необхідно здійснити сортування об'єктів за їх атрибутами, наприклад, за датою або іншими параметрами. Після сортування об'єкти можуть бути виведені в консоль або оброблені іншим чином, залежно від потреб програми.

Ці два етапи дозволяють розділити роботу на більш зручні компоненти, що полегшує процес розробки та збереження коду в майбутньому.

**План першого етапу:**

Нижче наведенi блок-cхеми роботи алгоритмiв iмпортування CSV файлу та переведення його в об’єкти класу Post, що дасть змогу полегшити роботу з сортуванням кожного запису або посту соцiальних мереж та обчислювати загальний критерiй зацiкавленостi аудиторiї.

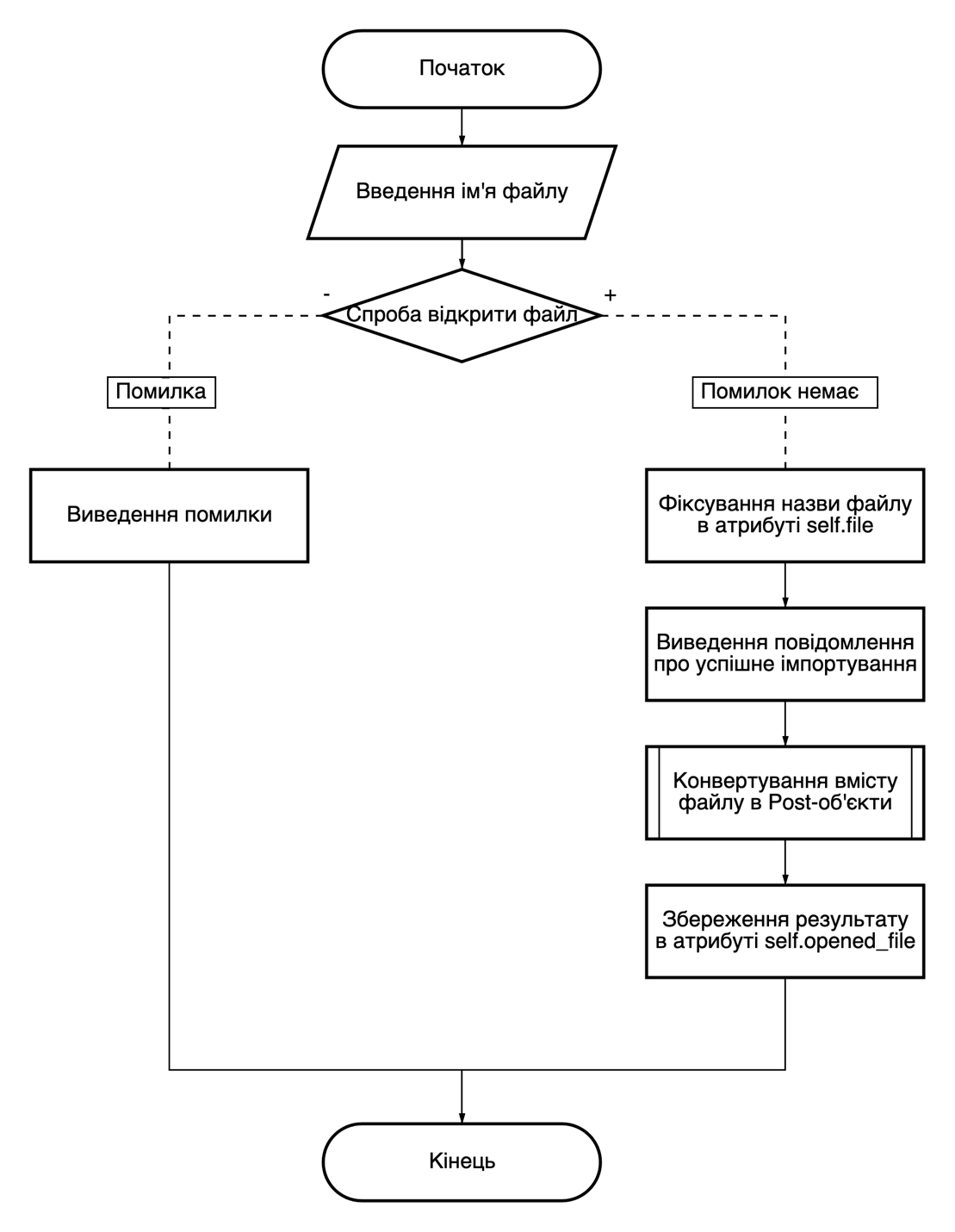
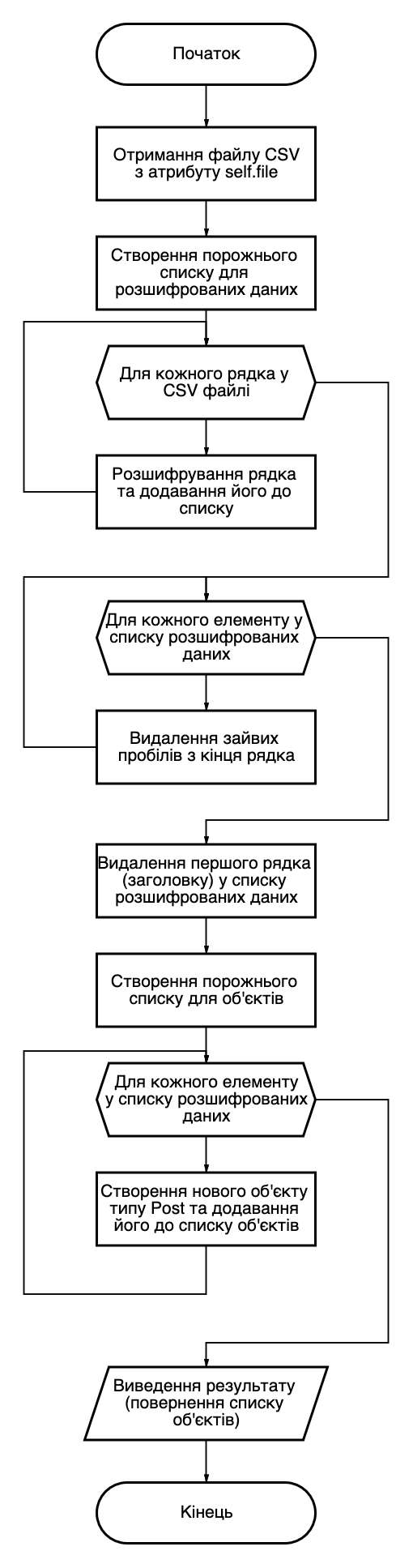


Рис. 2.1 – Схема функції iмпорту файлу

Рис. 2.2 – Схема методу переводу даних CSV файлу в об’єкти Post

**План другого етапу:**

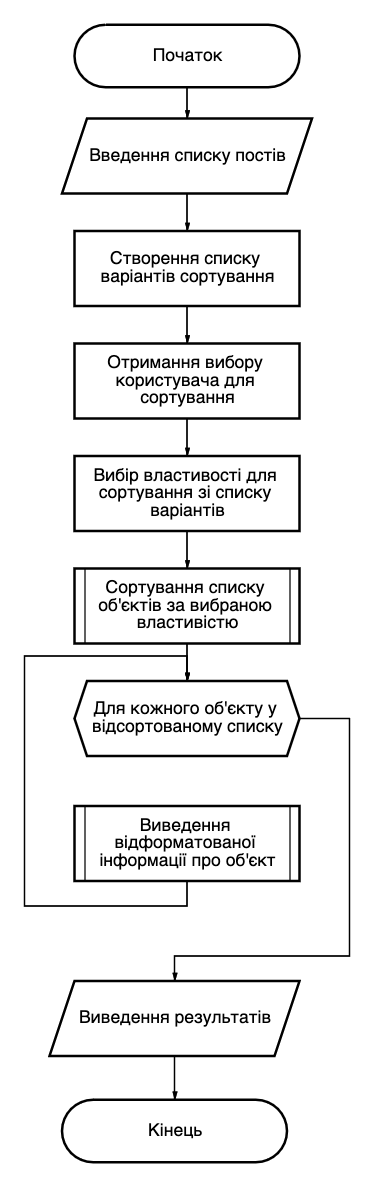
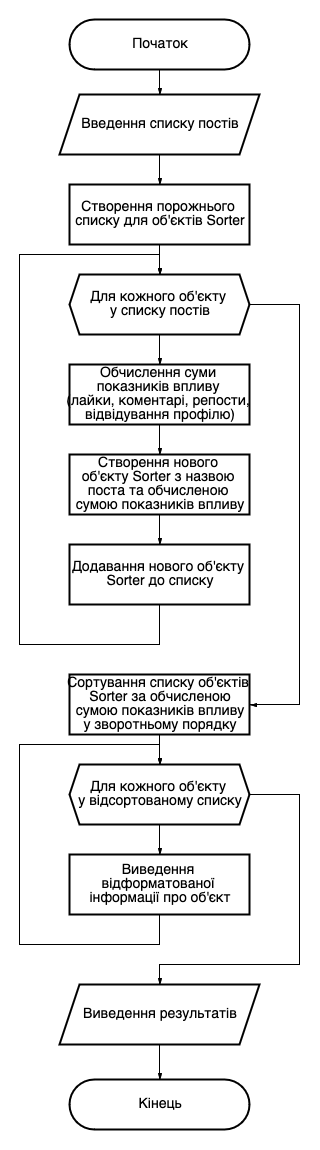
****Нижче наведенi блок-cхеми роботи алгоритмiв сортування об’єктiв класу Post та виведення загального значення популярностi посту, це допоможе мати бiльш краще уявлення про взаемодiю користувачiв з постом.

Рис. 2.3 – Схема методу сортування об’єктiв Post

****Рис. 2.4 – Схема методу узагальнення взаемодiй з постом – show\_popular

**3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

**3.1 Розробка головного класу Post**

Для створення програмного забезпечення з метою аналізу даних з соціальних медіа з CSV-файлу, який містить інформацію про публікації, лайки, коментарі та репости, потрібно виконати такі кроки:

1. **Визначення вимог до програми**:
   * Розробка функціоналу для аналізу популярності та взаємодії з контентом.
   * Опрацювання даних з CSV-файлу: визначення структури файлу, вибір необхідних полів для аналізу.
2. **Розробка архітектури програми**:
   * Визначення компонентів програми: модулі для обробки даних, аналізу популярності, взаємодії з контентом тощо.
   * Встановлення взаємозв'язків між компонентами, їх взаємодія для забезпечення потрібного функціоналу.
   * Вибір оптимальних методів обробки та аналізу даних для забезпечення швидкості та ефективності програми.
3. **Реалізація компонентів програми**:
   * Розробка коду компонентів відповідно до принципiв ООП.
   * Забезпечення коректної роботи кожного компонента та їх взаємодії з даними з CSV-файлу.
4. **Тестування програми**:
   * Проведення тестів для перевірки коректності роботи програми, відповідно до заданих вимог.
   * Виявлення та усунення будь-яких помилок або недоліків.
5. **Виправлення та перевірка правильності роботи**:
   * Внесення виправлень у програму, якщо необхідно після тестування.
   * Повторна перевірка програми для забезпечення її коректної роботи після внесених змін.

З вже визначеними вимогами та розробленою архiтектурою можна приступити до розробки програмного коду:

Для легкого манiпулювання даними що знаходяться в CSV файлi їх рекомендуеться перевести в об’єкти, для цього створюємо клас Post, який має всi необхiднi атрибути, такi як:

self.post\_title

self.likes

self.comments

self.reposts

self.profile\_visits

Рис. 3.1.1 – Класс об’єкту Post

**Методи**

**\_\_str\_\_:** Цей метод визначає рядкове представлення об'єкта типу "пост". Він викликається при використанні функції str() або при виведенні об'єкта на екран. У цьому випадку він повертає рядок, який містить інформацію про заголовок поста, кількість лайків, коментарів, репостів і відвідувань профілю.

**print\_out**: Цей метод створений для форматованого виведення інформації про об'єкт типу "пост" у терміналі з використанням кольорів, які імпортовані з класу Bcolors. Він повертає рядок, який містить інформацію про заголовок поста, кількість лайків, коментарів, репостів і відвідувань профілю, причому кожен параметр окремо забарвлюється відповідним кольором.

В даному прикладi у кожного атрибуту є параметри за замовчуванням, якi використовують клас PostTitle та Generator щоб створити начальний CSV файл для тестування програми, давайте розглянемо в пунктi 3.2:

**3.2 Класи для тестування**

Клас Generator має в собi атрибут rows(поля) та метод **generate()** який за кiлькiстю полiв створює випадковi об’єкти класу Post та записує їх в файл.



Рис. 3.2.1 – Клас Generator

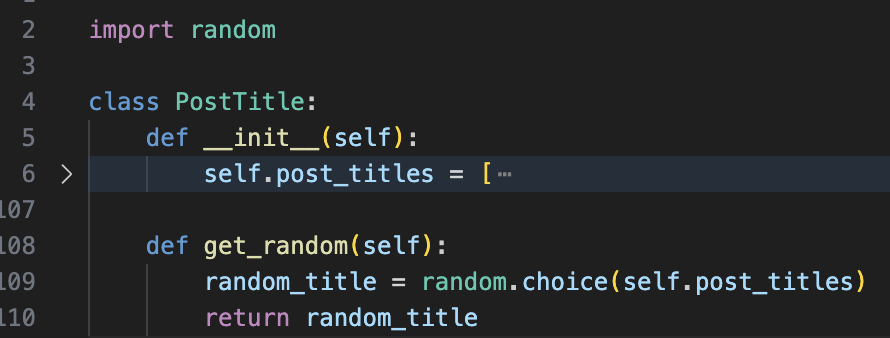
Клас PostTitle має список вже заданих назв постiв та метод який повертає один з цих назв – **get\_random()**.

Рис. 3.2.2 – Клас PostTitle

Всi цi класи були створенi для спрощення тестування програми i не мають прямого впливу на її роботу, хоча й дуже допоможуть в майбутньому, коли програма буде повнiстю готова.

**3.3 Клас FileManager**

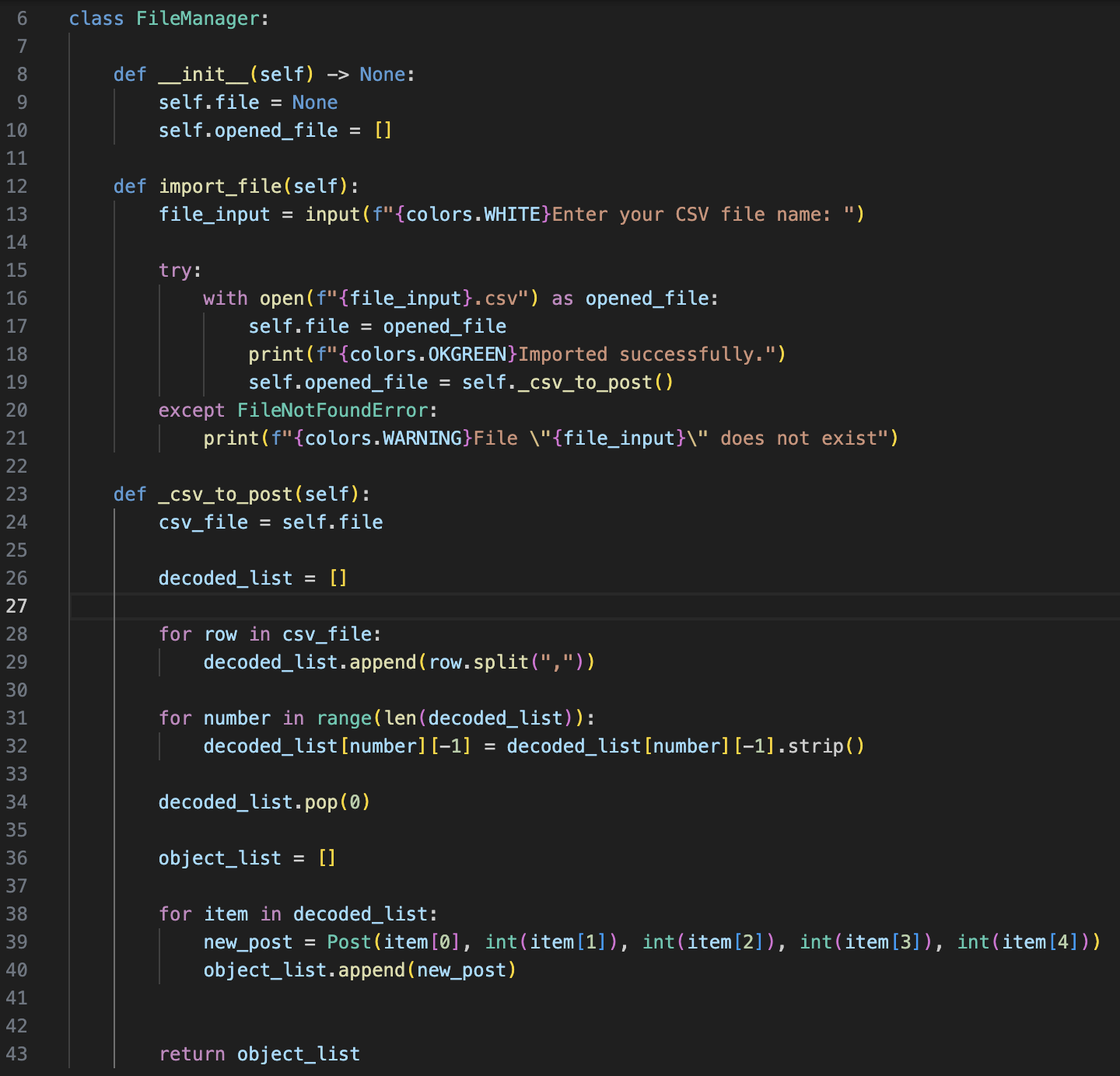
Перейдемо до створення класу який буде вiдповiдати за iмпортування самого CSV-файлу та переведенню його в об’єкти Post та буде зручно зберiгати їх в атрибутi opened\_file який є масивом.

Рис. 3.3.1 – Клас FileManager

**Принцип дiї:**

1. **Ініціалізація:** У конструкторі \_\_init\_\_, він ініціалізує дві змінні: file (яка представлятиме відкритий файл) та opened\_file (яка буде зберігати об'єкти, що відображають дані з CSV файлу).
2. **Імпорт файлу:** Метод import\_file отримує назву файлу від користувача, як введення. Він відкриває цей файл (якщо він існує), читає його вміст і перетворює його на список об'єктів Post. Якщо файл не знайдено, виводиться відповідне повідомлення.
3. **Перетворення CSV у пости:** Метод \_csv\_to\_post відповідає за перетворення вмісту CSV файлу на об'єкти Post. Він читає CSV файл, розділяє кожен рядок за комами, видаляє зайві пробіли, видаляє перший рядок (який зазвичай містить заголовки стовпців), і створює об'єкти Post з кожного рядка, використовуючи дані з цього рядка.

**3.4 Клас Bcolors**

Ще один клас який можна вiднести до допомiжних – Bcolors, ось що про нього потрiбно знати:

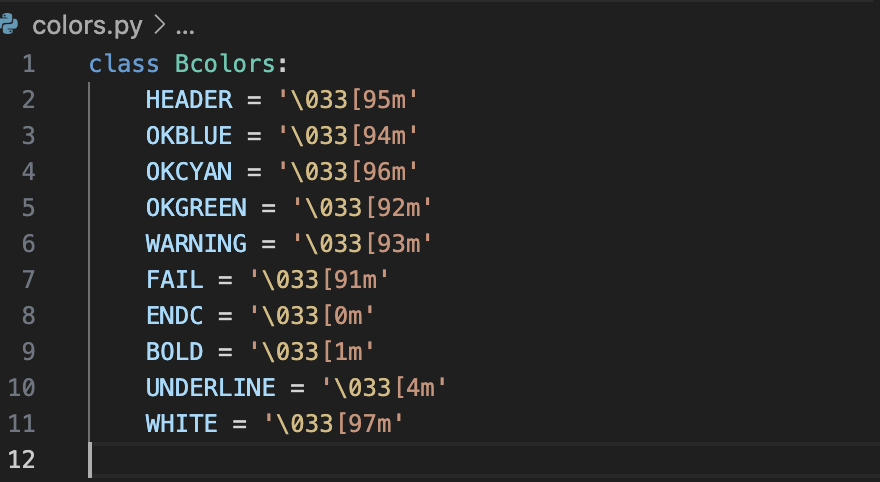
1. **Статичні атрибути:** Клас Bcolors має ряд статичних атрибутів, які представляють різні кольори. Кожен атрибут є рядком, що містить спеціальний код кольору ANSI для відображення тексту в відповідному кольорі у терміналі.
2. **Використання кольорів:** Ці атрибути можна використовувати, додавши їх до виведеного тексту за допомогою спеціальних кодів у форматі Escape-послідовностей. Наприклад, Bcolors.OKGREEN додасть код кольору, щоб текст виводився зеленим кольором. Це може бути корисно для покращення зручності виведення інформації в консольному інтерфейсі вашої програми.
3. **Нестатичні атрибути та конструктор:** Оскільки всі атрибути цього класу є статичними, йому не потрібен конструктор. Статичні атрибути доступні через ім'я класу, а не через екземпляр класу. Тому немає потреби в ініціалізації цих атрибутів для кожного об'єкта класу. Вони доступні всюди в програмі, де імпортований цей клас.

Рис. 3.4.1 – Клас Bcolors

**3.5 Сортування та клас Sorter**

Одним із найважливіших завдань у рамках цього проекту є організація та обчислення загальних показників для кожного поста. Для цього був створений окремий клас під назвою Sorter, який допомагає в сортуванні та аналізі інформації.

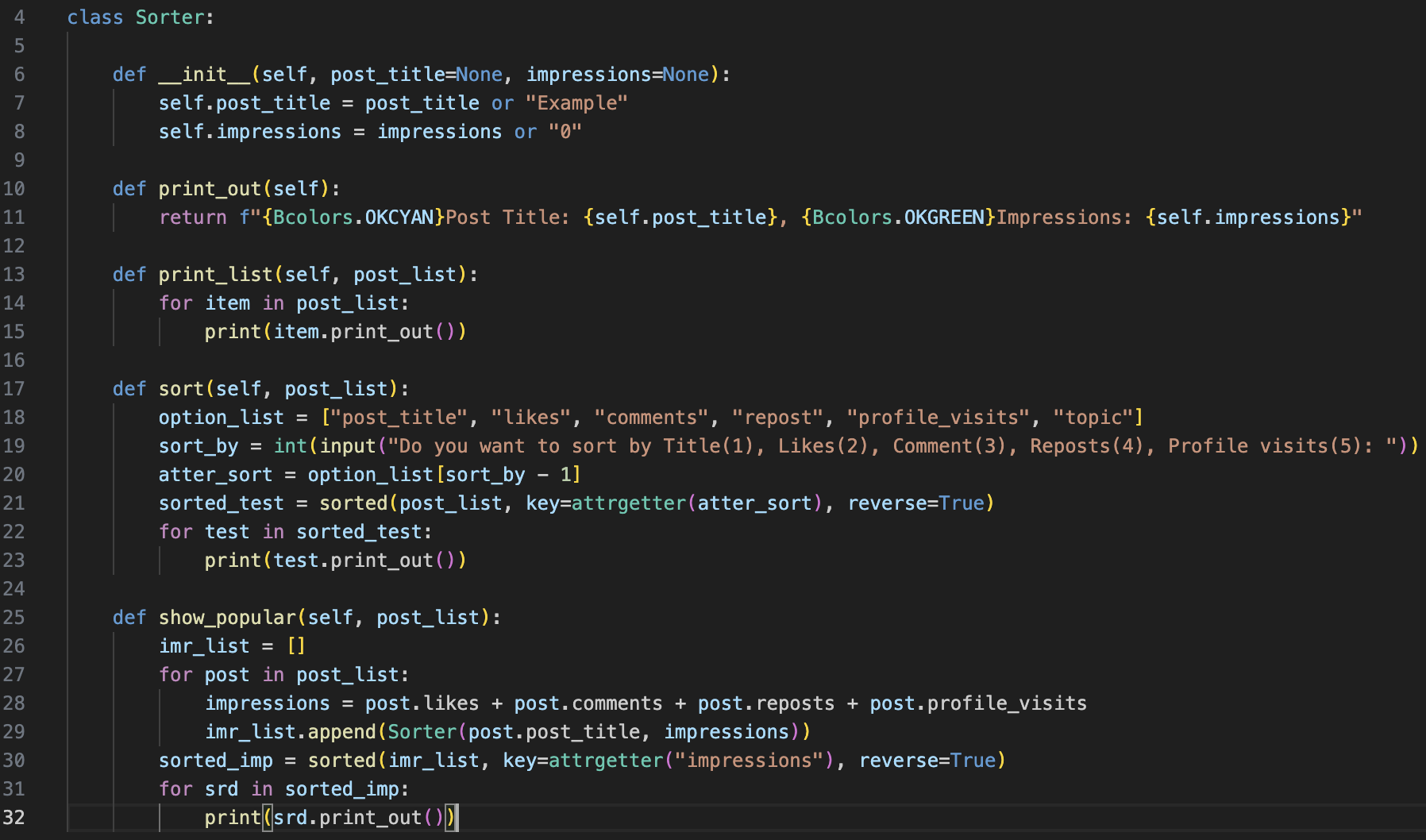
Клас Sorter має ряд методів, серед яких конструктор (**init**), який ініціалізує атрибути об'єктів, такі як назва поста (post\_title) та кількість переглядів (impressions). Метод print\_out виводить інформацію про пост, включаючи назву та кількість переглядів. Метод print\_list приймає список постів і виводить інформацію про кожен з них. Метод sort сортує список постів за вибраним параметром (наприклад, назвою, кількістю лайків тощо) та виводить відсортований список. Метод show\_popular аналізує популярність кожного поста, розраховуючи загальну кількість взаємодій (лайки, коментарі, репости, відвідування профілю) і виводить пости в порядку спадання популярності.

Рис. 3.5.1 – Клас Sorter

Цей клас був створений таким, як він є, для того, щоб забезпечити зручний та ефективний інструмент для обробки даних про пости у соціальних мережах. Основна мета цього класу полягає в тому, щоб упорядковувати та аналізувати інформацію про пости з різних кутів зору, таких як загальна популярність, кількість лайків, коментарів, репостів тощо.

Створення окремого класу дозволяє легко маніпулювати та аналізувати дані, використовуючи спеціалізовані методи та функції. Наприклад, метод sort дозволяє швидко сортувати список постів за різними критеріями, тоді як метод show\_popular робить можливим виділення найпопулярніших постів.

Крім того, використання класу дозволяє зберігати інформацію про кожен пост у відокремлених об'єктах, що спрощує доступ до цієї інформації та робить код більш структурованим та зрозумілим для подальшого розширення та модифікації. Такий підхід сприяє підтримці чистого та організованого коду, що є важливим для подальшого розвитку проекту та пiдтримання принципiв ООП.

**3.6 Основний файл виконання програми**

Файл main.py - це головний файл програми, де відбувається взаємодія з іншими частинами програми. У парадигмі об'єктно-орієнтованого програмування (ООП) main.py виконує роль координатора роботи об'єктів та визначає послідовність їх викликів та дій.

У моему коді, main.py відповідає за створення об'єктів Post, FileManager і Sorter, тобто за їх створення та управління.

Структура файлу main.py досить проста і пряма:

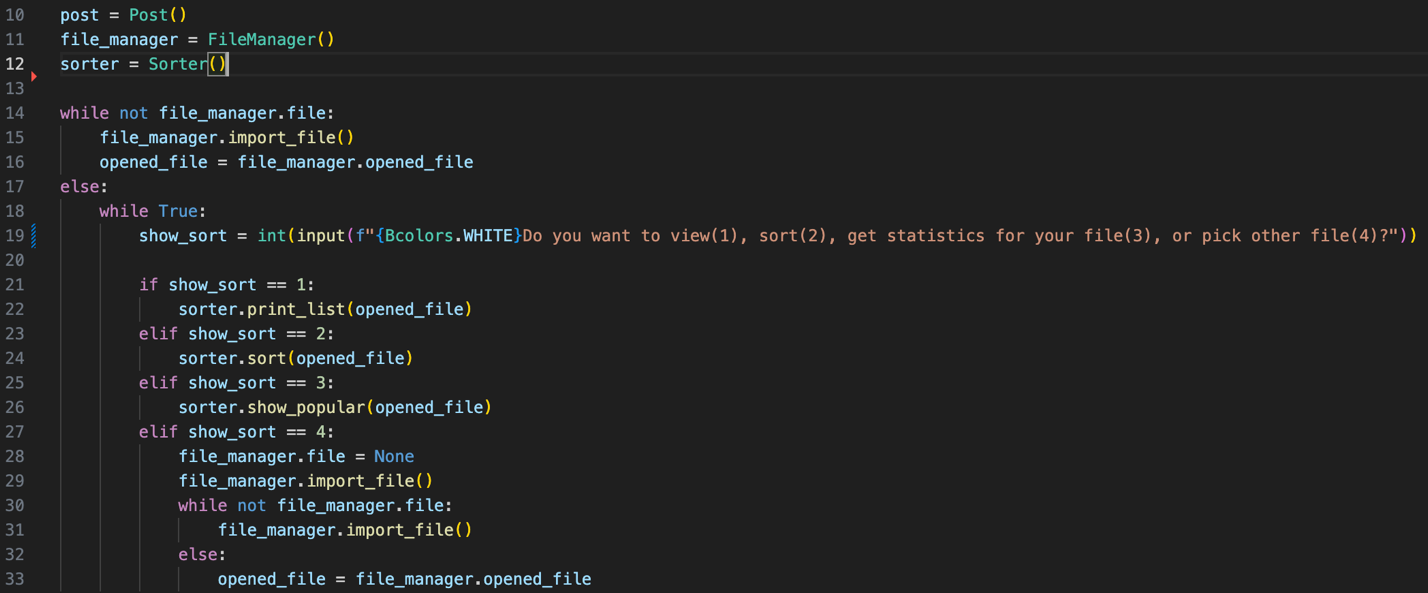
1. Спочатку створюються об'єкти для постів (Post), файлового менеджера (FileManager) та сортувальника (Sorter).
2. Потім main.py імпортує файл (з використанням методу import\_file() з файлового менеджера) та отримує відкритий файл.
3. Далі створюється цикл, який перевіряє, чи існує файл. Якщо файл не існує, програма намагається знову імпортувати його.
4. Після цього відбувається безкінечний цикл, де користувач може обирати, що саме він хоче зробити з файлом: переглянути, відсортувати, отримати статистику або вибрати інший файл. Залежно від вибору користувача, викликаються відповідні методи об'єкта Sorter, який здійснює відповідні операції над відкритим файлом.

Рис. 3.6.1 – Файл main.py

На цьому етапi розробку коду ПО можна вважати закiнченною, всi потрiбнi класи, методи та модулi були створенi та всi вимоги до функцiоналу програми задовiльненi.

1. **КЕРІВНИЦТВО КОРИСТУВАЧА**

**4.1 Основний функцiонал**

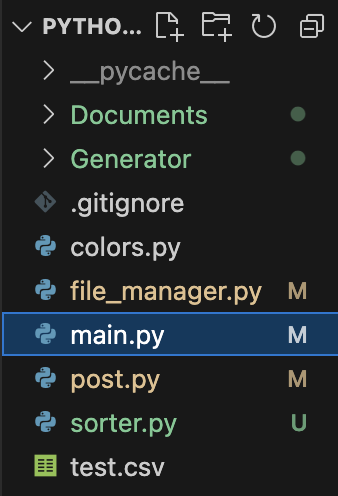
**Програма має наступний функцiонал:**

* Перегляд файлу в форматованому виглядi (view)
* Сортування за назвою (Title)
* Сортування за кiлькiстю лайкiв (Likes)
* Сортування за кiлькiстю коментарiв (Comment)
* Сортування за кiлькiстю репостiв (Reposts)
* Сортування за кiлькiстю вiзитiв профiлю з посту (Profile visits)
* Отримання загальної статистики для кожного посту (get statistics for your file)
* Вибiр iншого файлу (pick other file)

**4.2 Запуск програми**

Для запуску програми треба виконати наступнi дiї:

1. Перейти в внутрiшнiй каталог програми.
2. Вiдкрити файл main.py.
3. В Visual Studio Code, в правому верхньому кутку натиснути на кнопку “▶” (Run Pyhon File).

Рис. 4.2.1 – Файл main.py в кореневому каталозi

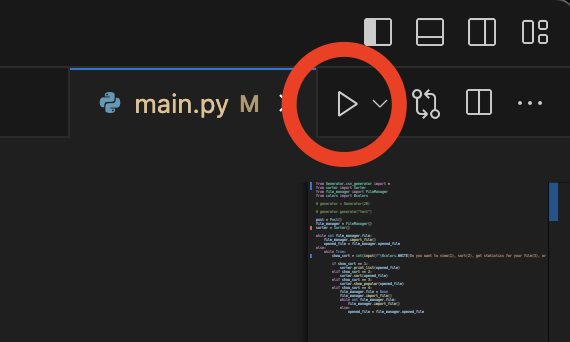
****

Рис. 4.2.2 – Кнопка запуску файлу (вiдмiчена червоним кольором)

**4.3 Iмпортування файлу**

Перед використанням програми треба переконатися в наявностi сумiсного CSV файлу в корневому каталозi, сумiсний файл повинен мати наступнi колонки в зазначеному порядку:

Post title, Likes , Comments, Reposts, Profile visits

1. Пiсля переконання в тому, що файл мiститься в каталозi програми, запустiть програму:

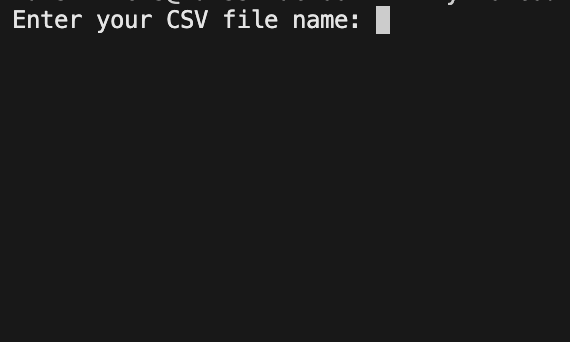


Рис. 4.3.1 – Початковий екран вибору файлу

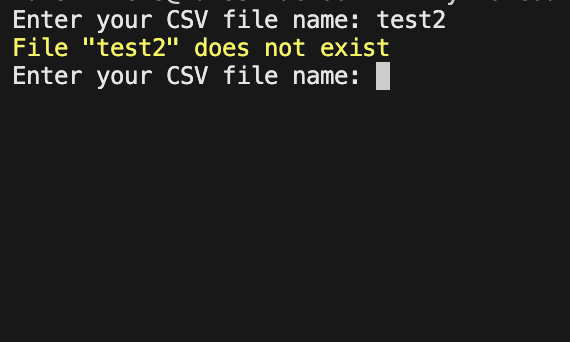
2. Пicля запуску, на початковому екранi (Рис. 4.3.1), введiть iм’я файлу у форматi лише назви (без формату файла).

Рис. 4.3.2 – Помилка при знаходженнi файлу

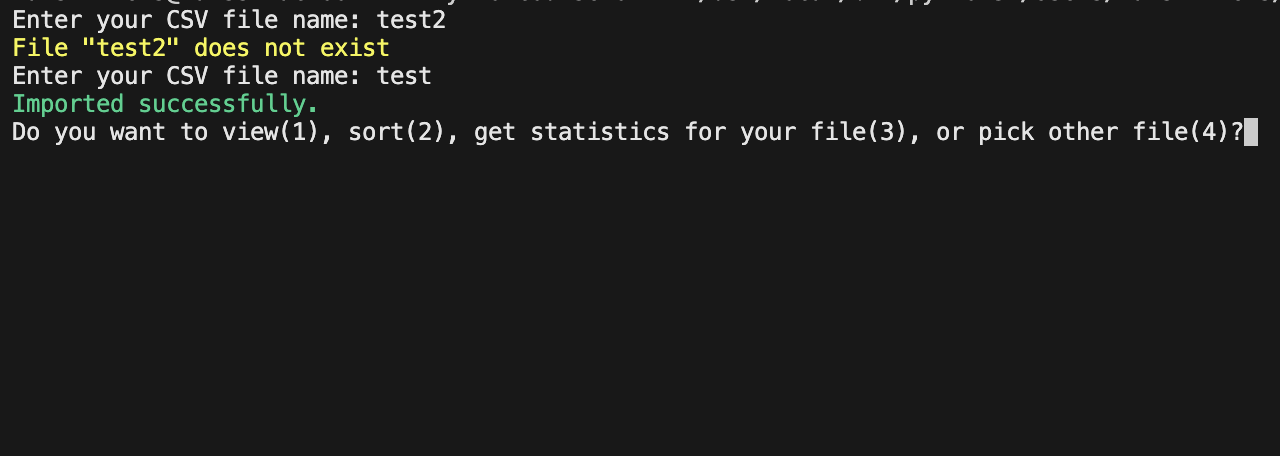
3. Помилка “File “**iм’я файлу**” does not exist” (Рис 4.3.2) наголошує користувачевi на тому, що файлу з введеною назвою не iснує, в такому випадку рекомендується перевiрити наявнiсть файлу в корневiй папцi програми або впевнитися що назва була вказана правилльно.

Рис. 4.3.3 – Приклад коректно введеного файлу

1. Якщо файл було введено коректно програма засвiдчить про це написом “Imported successfully” (Рис 4.3.3)

**4.4 Робота з файлом**

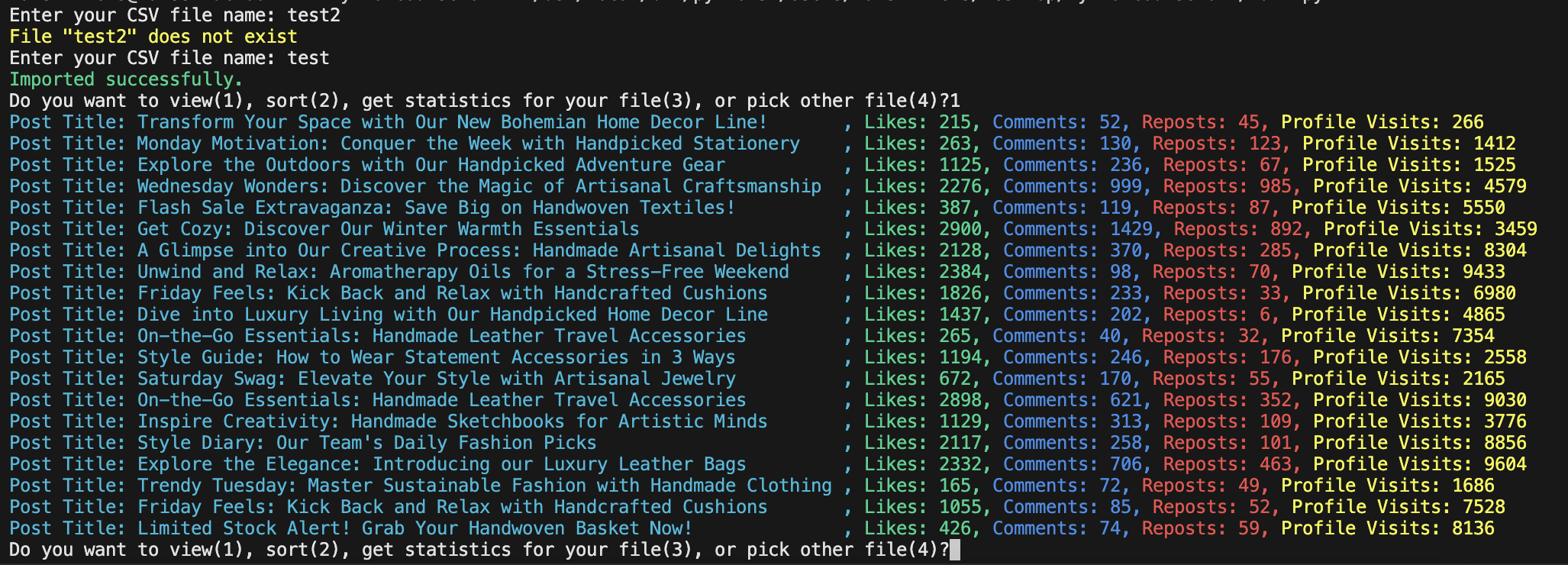


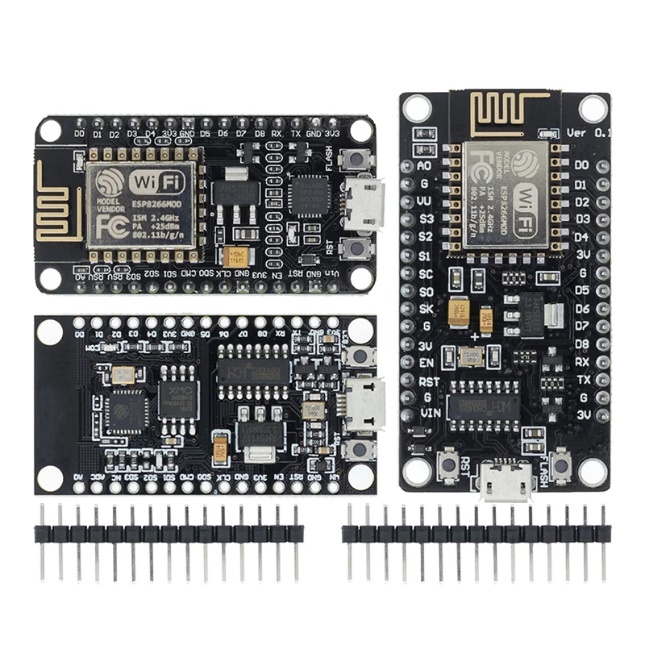
Рис. 4.3.4 – Перегляд файлу за допомогою опцiї “view”

2. Аналітичний огляд мікроконтролерної плати Node MCU

2.1 NodeMCU

NodeMCU — це недорога IoT платформа з відкритим кодом. Спочатку він включав мікропрограмне забезпечення, яке працює на ESP8266 Wi-Fi SoC від Espressif Systems, і апаратне забезпечення на основі модуля ESP-12. Пізніше була додана підтримка 32-розрядного MCU ESP32.

NodeMCU може використовуватися як для створення автономних інтерактивних об'єктів, так і підключатися до програмного забезпечення, яке виконується на комп'ютері (наприклад: Processing, Adobe Flash, Max/MSP, Pure Data, SuperCollider). Інформація про плату (рисунок друкованої плати, специфікації елементів, програмне забезпечення) знаходяться у відкритому доступі і можуть бути використані тими, хто воліє створювати плати власноруч.

Рис. 2.1.1 - NodeMCU

2.2 Розміри і габарити плати

Розмір плати NodeMCU – 6\*3 см. Плата досить компактна, це дозволяє використовувати її у більшій кількості проектів.

2.3 Живлення

NodeMCU може отримувати харчування через підключення USB або від зовнішнього джерела живлення. Джерело живлення вибирається автоматично.

Зовнішнє живлення (НЕ USB) може подаватися через перетворювач напруги AC / DC (блок живлення) або акумуляторною батареєю. Перетворювач напруги підключається за допомогою роз'єму 2.1 мм з центральним позитивним полюсом. Провід від батареї підключаються до виводів Gnd і Vin роз'єму живлення.

Платформа може працювати при зовнішньому живленні від 6 В до 20 В. При напрузі живлення нижче 7 В, висновок 5V може видавати менше 5 В, при цьому платформа може працювати нестабільно. При використанні напруги вище 12 В регулятор напруги може перегрітися і пошкодити плату. Рекомендований діапазон від 7 В до 12 В.

2.4 Входи і виходи

З використанням функцій pinMode (), digitalWrite () і digitalRead () кожен з 14 цифрових виводів може працювати в якості входу або виходу. Рівень напруги на виводах обмежений 5 В. Максимальний струм, який може віддавати або споживати один вивід, становить 40 мА. Всі виводи пов'язані з внутрішніми підтягуючими резисторами (за умовчанням відключеними) номіналом 20-50 кОм. Крім цього, деякі виводи Аrduino можуть виконувати додаткові функції:

Висновки живлення – на платі розташовано чотири виводи живлення, а саме: один висновок VIN та три виводи 3.3V. Якщо у вас є стабілізоване джерело напруги 5 В, висновок VIN можна використовувати для безпосереднього живлення ESP8266 та його периферії. Висновки 3.3V – це виходи вбудованого стабілізатора напруги. Ці висновки можуть бути використані для подачі живлення на зовнішні компоненти.

GND – це виведення землі налагоджувальної плати ESP8266 NodeMCU.

Виводи I2C використовуються для підключення всіх видів датчиків та периферійних пристроїв на шині I2C у вашому проекті. Підтримуються і I2C Master, і I2C Slave. Робота інтерфейсу I2C можна реалізувати програмно, а тактова частота становить максимум 100 кГц. Слід зазначити, що тактова частота I2C повинна бути вищою за найнижчу тактову частоту з ведених пристроїв.

Висновки GPIO ESP8266 NodeMCU має 17 висновків GPIO, які можна призначати програмно на різні функції, такі як I2C, I2S, UART, PWM, дистанційне інфрачервоне управління, світлодіодний індикатор і кнопка. Кожен увімкнений висновок GPIO може бути налаштований або на внутрішню підтяжку до землі або до шини живлення, або встановлений на високоімпедансний стан. При конфігуруванні на вхід для генерування переривань процесора він може бути налаштований на спрацьовування або фронтом, або спадом.

Висновок ADC подає сигнал на наявний NodeMCU, вбудований 10-розрядний прецизійний аналого-цифровий перетворювач послідовного наближення (SAR ADC). За допомогою цього АЦП можуть бути реалізовані дві функції: перевірка напруги живлення на виводі VDD3P3 та перевірка вхідної напруги на виводі TOUT (але не одночасно).

Висновки UART ESP8266 NodeMCU має 2 інтерфейси UART, тобто UART0 та UART1, які забезпечують асинхронний зв'язок (RS232 та RS485) та можуть обмінюватися даними зі швидкістю до 4,5 Мбіт/с. Для зв'язку можна використовувати UART0 (висновки TXD0, RXD0, RST0 та CTS0), який підтримує керування потоком. UART1 (висновок TXD1) підтримує лише сигнал передачі, тому він зазвичай використовується для друку журналу подій.

Висновки SPI ESP8266 має два інтерфейси SPI (SPI та HSPI), що підтримують і ведений (slave), і ведучий (master) режими. Ці інтерфейси SPI також підтримують такі функції SPI:

4 режими синхронізації передачі SPI;

до 80 МГц та тактові частоти, отримані розподілом 80 МГц;

до 64 байт FIFO.

Виводи SDIO ESP8266 має захищений цифровий інтерфейс вводу/виводу (SDIO, Secure Digital Input/Output Interface), який використовується для прямого підключення карток SD. Підтримуються 4-бітний 25 МГц SDIO v1.1 та 4-бітний 50 МГц SDIO v2.0.

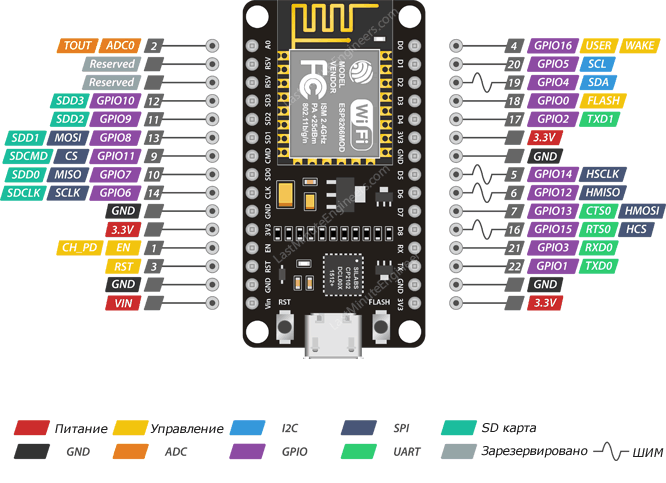
Висновки PWM На платі є 4 канали широтно-імпульсної модуляції (PWM). Вихід ШІМ може бути реалізований програмно та використаний для управління двигунами та світлодіодами. Частотний діапазон ШІМ регулюється від 1000 до 10000 мкс, тобто від 100 Гц до 1 кГц.

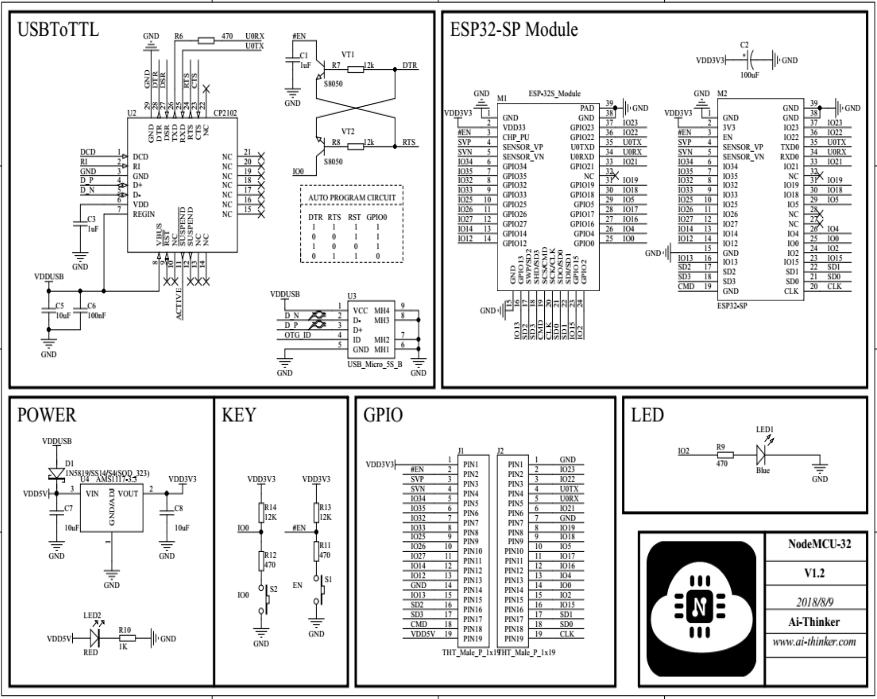
Висновки керування використовуються, як не дивно, для керування ESP8266. Ці висновки включають висновок включення мікросхеми EN, висновок скидання RST і виведення пробудження WAKE.

Висновок EN – мікросхема ESP8266 включена, коли виведення EN подається високий логічний рівень. За низького логічного рівня мікросхема працює на мінімальній потужності.

Висновок RST використовується для скидання мікросхем ESP8266.

Виведення WAKE використовується для виведення чіпа із глибокого сну.

Рис. 2.4.1 — Node MCU – призначення пінів



,

Рис 2.4.2 - Принципова схема плати Arduino Uno R3

3. Огляд вхідних та вихідних ланцюгів

3.1 Огляд P10 RGB Matrix дiсплея.

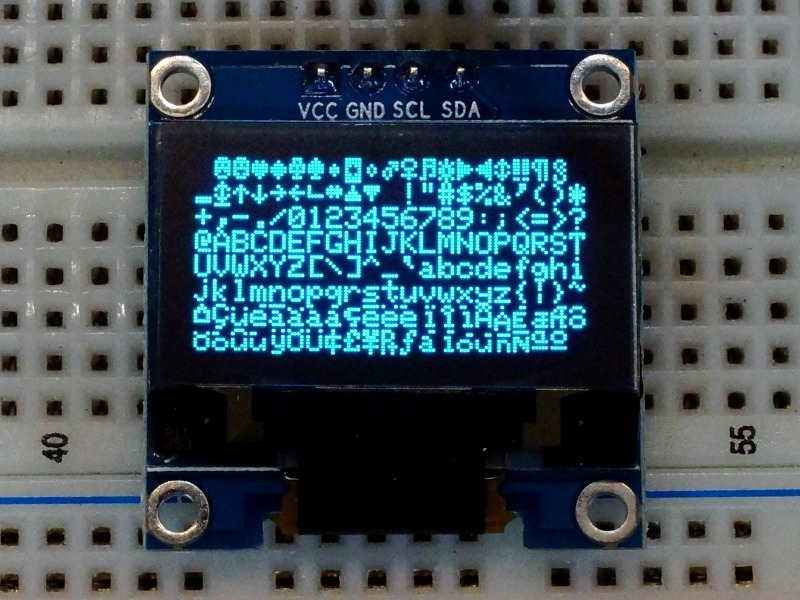
Дуже якісний дисплей OLED з високою контрастністю. Легко підключається, швидкодія, можливе підключення відразу декількох дисплеїв до будь-якого мікроконтролера або міні-комп'ютера по шині I2C. Адреса можна вибрати перепаюванням перемички. За замовчуванням вибрано адресу 0x3c.

GND: Загальний

VCC: Напруга живлення

SDA: Шина даних

SCL: Шина тактування даних

Рис 3.1.1 — OLED I2C 128x64 0.96

3.3 Технічні характеристики дисплея

* Роз'єм 4-pin
* напруга живлення 3.3-6 -volt
* Роздільна здатність дисплея 128\*64
* Розмір екрана 0.96"
* Тип екрану OLED
* Товщина 11 мм
* Ширина 27 мм
* Висота 27 мм
* Вага 4.0 грам

Специфікація OLED I2C 128x64:

* Напруга живлення: 3.3-6V
* Драйвер OLED модуля: SSD1306
* Кут огляду: > 160 градусів
* Рівні вхідних сигналів: 3.3V/5V
* Колір: опція вибору - синій/білий/синій+жовтий

3.4 Датчик DHT11

**Датчик DHT11** – це цифровий датчик температури та вологості, що дозволяє калібрувати цифровий сигнал на виході. Складається з ємнісного датчика вологості та термістора. Також, датчик містить в собі АЦП для перетворення аналогових значень вологості та температури.

Мінімальний розмір, низьке енергоспоживання датчика дозволяє використовувати його в мобільних пристроях. Модуль може монтуватися як на друковану плату, так і підключатися за допомогою стандартних дротів.

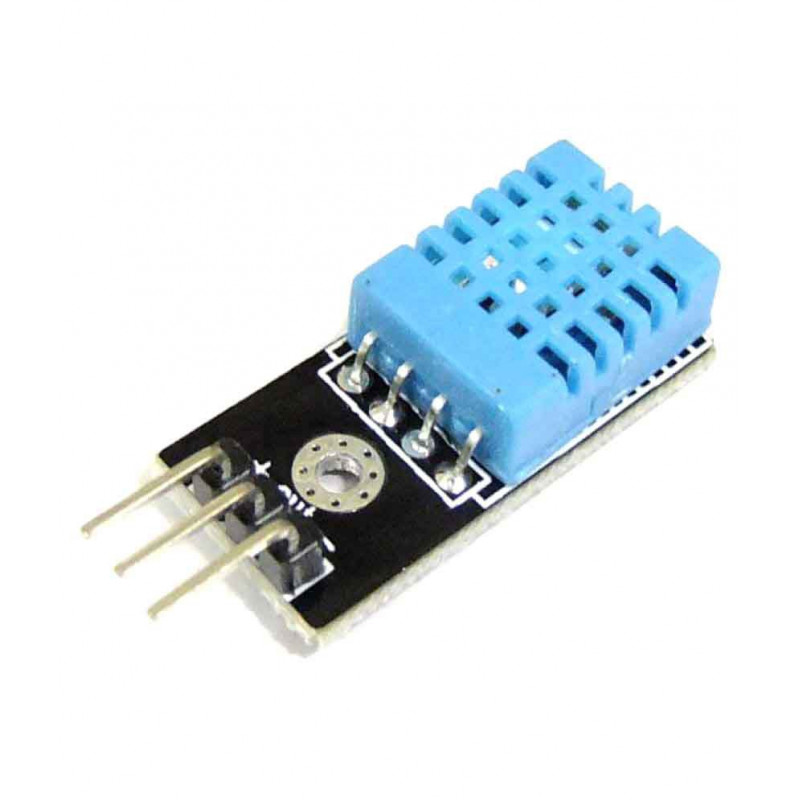


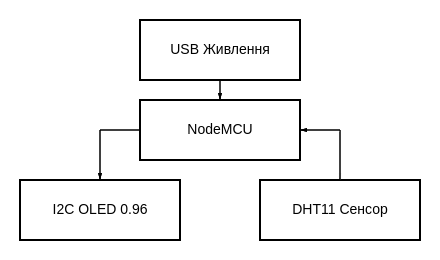
Рис. 3.4.2 - Датчик DHT11

3.5 Технічні характеристики DHT11

Модель виробника: ASAIR DHT11

* Визначення вологості: 5 - 95% RH ± 5% (макс.)
* Визначення температури: -20 ~ +60 ºC ± 2% (макс.)
* Живлення: 3.5-5.5 В
* Частота опитування: не більше 1 Гц
* Розміри 15.5 x 12 x 5.5 мм
* **Пiдсумки:**  
  1.VCC(3-5вживлення)  
  2.DataOut-вивідданих  
  3.NC-невикористовується  
  4. Загальний

4. Розробка структурної схеми мікропроцесорної системи



5. Розробка принципової схеми мікропроцесорної системи

5.1 Принципова електрична схема

Електрична принципова схема (ДСТУ ГОСТ 2.702:2013) є видом електричної схеми виробу, що дає найповніше уявлення про склад і принцип його роботи. При виконанні схем цифрової обчислювальної техніки керуються ГОСТ 2.708-81. Цей вид кресленика не враховує габаритних розмірів і реального розташування деталей об'єкта. За рівнем абстракції принципові електричні схеми займають середню позицію між функціональними і монтажними схемами і позначаються у шифрі основного напису символами *Э3*.

Принципові електричні схеми призначенні для повного відображення взаємозв'язків пристроїв з урахуванням принципів їх дії і послідовності роботи. На принципових електричних схемах електричні елементи зображують за допомогою умовних позначень, а також вказують лінії зв'язків між ними, блоками та модулями. На схемі, також, розміщується наступна інформація: умовне зображення принципу роботи функціональних вузлів, пояснювальні написи, частини окремих елементів, тощо.

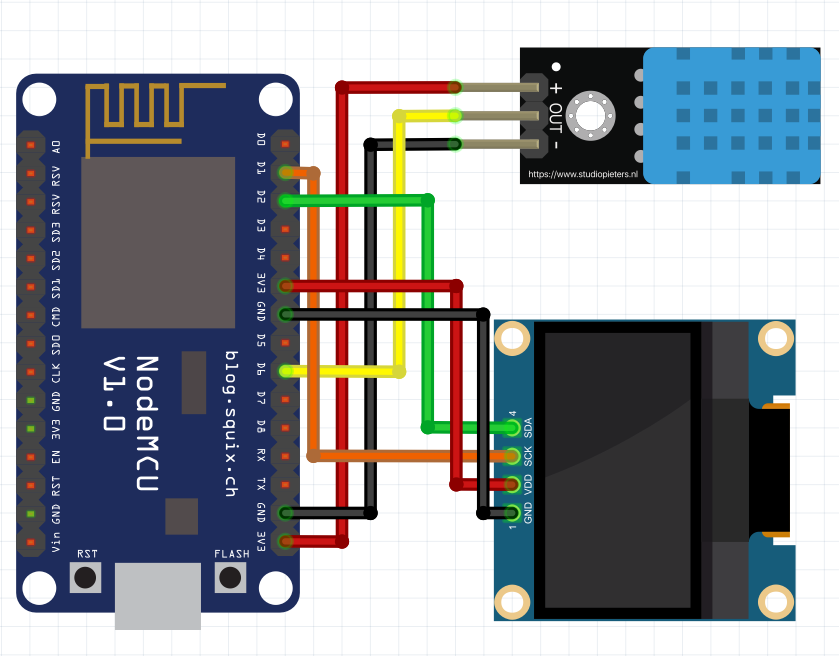
5.2 Fritzing

Fritzing — програмне забезпечення з відкритим кодом, орієнтоване на розробку схем та друкованих плат для Arduino.

Для створення принципової схеми мікропроцесорної системи контролю погоди потрібно:

* Node MCU v3
* Макетна плата
* Дисплей **OLED I2C 128x64 0.96**
* **Датчик BMP180**

Принципова схема мікропроцесорної системи зображена на рис. 5.2.1 та рис. 5.2.2

Рис. 5.2.1 – Принципова схема

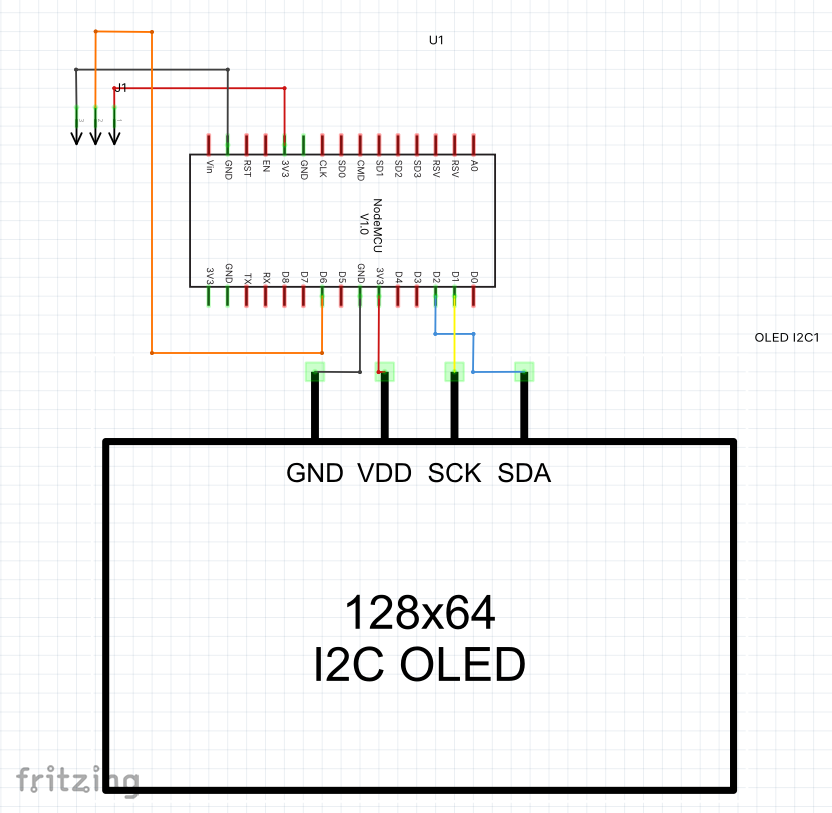
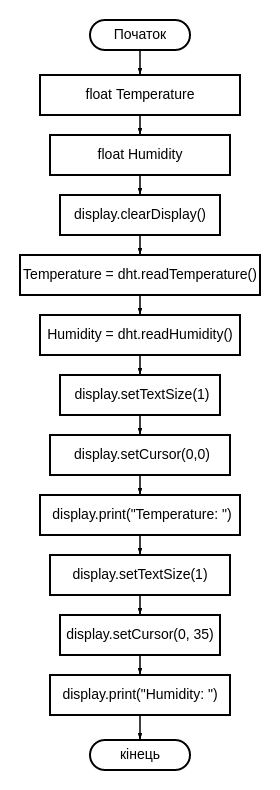


Рис. 5.2.2 – Принципова схема

|  |  |  |  |
| --- | --- | --- | --- |
| No | Позначення | Назва деталi | Кiлькiсть |
| 1 | Node MCU | Плата ESP8266 | 1 |
| 2 | DHT11 | Датчик температури та вологостi | 1 |
| 3 | I2C OLED 0.96 | Дисплей | 1 |
| 4 | Mod1 | З’еднувальний провiд | 7 |

6. Розробка алгоритму роботи мікропроцесорної системи



7. Розробка програмного забезпечення для мікропроцесорної системи

#include <Wire.h>

#include "DHT.h"

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h> //бiблiотеки

Adafruit\_SSD1306 display = Adafruit\_SSD1306(128, 64, &Wire, -1);

unsigned long delayTime;

#define DHTTYPE DHT11 // DHT 11

//DHT11 Сенсор;

uint8\_t DHTPin = 12;

DHT dht(DHTPin, DHTTYPE);

float Temperature;

float Humidity;

float Temp\_Fahrenheit;

void setup() {

Serial.begin(115200);

pinMode(DHTPin, INPUT);

dht.begin();

display.begin(SSD1306\_SWITCHCAPVCC, 0x3C); // initialize with the I2C addr 0x3C (for the 128x64) //iнiцiалiзацiя дiсплея

display.display();

delay(100);

display.clearDisplay();

display.display();

display.setTextSize(1.75);

display.setTextColor(WHITE);

}

void loop() { //Зчитування

Humidity = dht.readHumidity();

Temperature = dht.readTemperature();

Temp\_Fahrenheit= dht.readTemperature(true);

if (isnan(Humidity) || isnan(Temperature) || isnan(Temp\_Fahrenheit)) {

Serial.println(F("Failed to read from DHT sensor!"));

return;

}

//Виведення

Serial.print(F("Humidity: "));

Serial.print(Humidity);

Serial.print(F("% Temperature: "));

Serial.print(Temperature);

Serial.print(F("°C "));

Serial.print(Temp\_Fahrenheit);

Serial.println(F("°F "));

display.setCursor(0,0);

display.clearDisplay();

display.setTextSize(1);

display.setCursor(0,0);

display.print("Temperature: ");

display.setTextSize(2);

display.setCursor(0,10);

display.print(Temperature);

display.print(" ");

display.setTextSize(1);

display.cp437(true);

display.write(167);

display.setTextSize(2);

display.print("C");

// Temperature

display.setTextSize(1);

display.setCursor(0, 35);

display.print("Humidity: ");

display.setTextSize(2);

display.setCursor(0, 45);

display.print(Humidity);

display.print(" %");

display.display();

delay(1000);

Висновок

Портативнi системи контролю погоди дуже потрібні, адже вони показують найточніші дані для конкретного розташування та можуть бути використанi всюди. Схема проекту може працювати навiть вiд порту зарядки телефону або бути пiдключена до любих iнших зовнiшнiх батарей через USB з напругою 5V.

Застосування мікроконтролерів у техніці дуже актуальним та простим. Тому що вони суттєво прискорюють роботу поставленого їм завдання. Звідси і важливість їх вивчення та застосування у пристроях.

В данному курсовому проектi розглянув таку тему як: Розробка мікропроцесорної системи портативна метеостанція.

Во1 розділі було розглянуто що таке мікроконтролер, загальні відомості про RISC або СІSK, та засоби розробки.

Во2орозділі було розглянуто що таке NodeMCU, його характеристики та інше.

Во3орозділі було зроблено підбір та опис компонентів для мікропроцесорної системи метеостанцiї.

Во4 розділі було зроблено структурну схему мікропроцесорної системи.

Во5 розділі було розроблено принципову схему в додатку Fritzing.

Во6 розділі було розроблено алгоритм мікропроцесорної системи.

Во7 розділі було розроблено програмне забезпечення.

Перелік використаних джерел

1. ARDUINO від азів програмування до створення практичних приладів. Видавництво: Наука і техніка Автор: Белов А.В. 2018г.
2. https://aliexpress.com
3. https://ua.wikipedia.org

# ESP8266 Home Automation Projects Автор: Catalin Batrinu

1. 6. https://ekotechnik.ua
2. ДСТУ 2383-94 Блоки оброблення даних. Терміни та визначення