

**Міністерство освіти і науки України**  
**Чернівецький національний університет**  
**імені Юрія Федьковича**

Навчально-науковий інститут фізико-технічних та комп'ютерних наук

(повна назва інституту/факультету)

Відділ комп'ютерних технологій

Кафедра комп'ютерних систем та мереж

(повна назва кафедри)

**Звіт**  
**з переддипломної практики**

**Спеціальність 123 – Комп'ютерна інженерія**  
**Рівень вищої освіти – перший (бакалаврський)**  
**ОПП «Комп'ютерна інженерія»**

Керівник практики

докт. фіз.-мат. наук, доцент

\_\_\_\_\_ С.В. Баловсяк

“ \_\_\_ ” \_\_\_\_\_ 202\_ р.

Виконавець

студент 4-го курсу групи 442А

\_\_\_\_\_ І.Ю. Олександров

“ \_\_\_ ” \_\_\_\_\_ 202\_ р.

Чернівці – 202\_

**ЗМІСТ**

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ .....	3
ВСТУП .....	4
1. ПОСТАНОВКА ЗАВДАННЯ ТА ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ.....	5
1.1. Актуальність розробки комп'ютерної системи для аналізу значень температури і вологості .....	5
1.2. Опис предметної області та сфери використання .....	6
1.3. Аналіз комп'ютерних систем-аналогів для зчитування сигналів із сенсорів .....	7
1.4. Постановка завдання, опис вимог до системи на основі аналізу аналогів .....	10
2. ОГЛЯД ПІДХОДІВ ДО ВИРІШЕННЯ ЗАВДАННЯ .....	11
2.1. Загальна будова комп'ютерної системи для аналізу сигналів температури і вологості .....	11
2.2. Основи кластеризації даних .....	12
2.3. Функції комп'ютерної системи .....	17
3. ПРОЕКТУВАННЯ КОМП'ЮТЕРНОЇ СИСТЕМИ .....	18
3.1. Кластеризація методом k-середніх .....	18
3.2. Апаратно-програмна реалізація комп'ютерної системи .....	19
ВИСНОВКИ .....	27
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	28

**СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ**

Delta\_t – інтервал дискретизації (час між зчитуваннями сигналу).

GPIO – General-Purpose input/output (інтерфейс введення виведення загального призначення, через який можливе під'єднання сенсорів).

k – кількість кластерів.

QP – кількість значень сигналу.

$\rho$  – відстань між об'єктами.

ОС – операційна система.

## ВСТУП

Метою переддипломної практики є проведення наукових досліджень за тематикою розробки комп'ютерної системи для зчитування та аналізу сигналів із сенсорів температури і вологості, а також апаратно-програмна реалізація такої комп'ютерної системи.

Для досягнення мети практики використовуються такі методи дослідження: метод k-середніх для кластеризації зчитаних із сенсора значень температури та вологості, методи роботи з базою даних SQLite.

Завдання переддипломної практики наступні:

1. Аналіз будови і принципів роботи сенсорів температури та вологості, а також мікрокомп'ютера Raspberry Pi3.
2. Розробка апаратно-програмного забезпечення комп'ютерної системи для аналізу сигналів із сенсорів температури та вологості, тестування розробленої системи.

## 1. ПОСТАНОВКА ЗАВДАННЯ ТА ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1. Актуальність розробки комп'ютерної системи для аналізу значень температури і вологості

Аналіз значень температури та вологості є важливим для багатьох прикладних задач, наприклад, для контролю мікроклімату в житлових і виробничих приміщеннях. Проте, для подібних комп'ютерних систем контролю важливо мінімізувати їх розміри і масу, що технічно складно зробити при використанні окремих сенсорів температури і вологості. Тому в даній роботі значення температури і вологості зчитуються з одного сенсора DHT22 [1-2] за допомогою мікрокомп'ютера Raspberry Pi3 [3-4]. Це дозволило створити компактну систему з низькою споживаною потужністю.

Програмне забезпечення комп'ютерної системи для зчитування та аналізу сигналів із сенсорів розроблено на мові Python. Програму для зчитування сигналів із сенсорів розроблено в середовищі Thonny, а програму для аналізу отриманих сигналів розроблено на хмарній платформі GoogleColab. Завдяки цьому програму для аналізу отриманих сигналів можливо виконувати як за допомогою мікрокомп'ютера Raspberry Pi3, так і на хмарній платформі GoogleColab. Аналіз сигналів із сенсорів полягає у перевірці допустимості їх значень, а також у кластеризації отриманих значень температури та вологості методом k-середніх [5-7]. Завдяки такій обробці можливо виділяти як кластери характерні комбінації температури та вологості, що є важливим для контролю мікроклімату в приміщеннях. Отримані значення сигналів з сенсорів, параметри кластерів та інформація про місце та дату досліджень зберігаються в базі даних SQLite.

Тому під час практики проведено дослідження комп'ютерних систем для аналізу сигналів із сенсорів температури та вологості, а також розроблено апаратно-програмні засоби такої системи.

## 1.2. Опис предметної області та сфери використання

Предметною областю дослідження є зчитування та аналіз сигналів із сенсорів температури та вологості.

Сферою використання комп'ютерні системи для контролю значень температури та вологості.

Сенсори температури та вологості виконуються або як окремі пристрої, або ж інтегруються в одному пристрої. Розрізняють такі чотири типи сенсорів температури [8]:

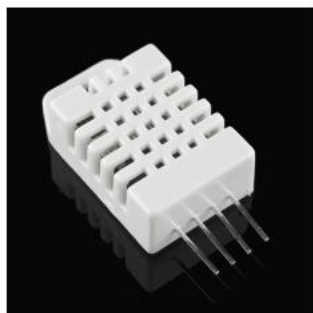
- 1) термоелементи (термопари);
- 2) резистивні детектори температури або терморезистори;
- 3) термістори;
- 4) напівпровідникові сенсори температури.

За будовою термопара – це два провідника, виготовлені з різних металів або сплавів, які з'єднуються (спаюються або зварюються) в одній точці. Принцип роботи термопар заснована на термоелектричному ефекті, тобто на чутливості вихідної напруги термопари до температури. В термопарах застосовуються певні з'єднання двох матеріалів, наприклад, міді і мідно-нікелевого сплавів, заліза і мідно-нікелевого сплавів. Перевагою термопар є можливість роботи при високих температурах, а недоліком є низька чутливість.

Принцип роботи терморезисторів заснований на зміні опору провідників залежно від зміни температури (для металів при підвищенні температури опір пропорційно збільшується). При використанні в терморезисторах нікелю чи платини можливе вимірювання температури в діапазоні від  $-200^{\circ}\text{C}$  до  $+850^{\circ}\text{C}$ .

Термістор відрізняється від терморезистора тим, що в ньому як провідний матеріал використовується не метал, а напівпровідник. Тому опір термістора при зростанні температури нелінійно зменшується. Як матеріали для термісторів використовуються суміші сульфідів і селенідів. Перевагою





**Standard AM2302/DHT22**



**AM2302/DHT22 with big case and wires**

### **Digital relative humidity & temperature sensor AM2302/DHT22**

а)

б)

Рисунок 1.2 – Сенсор температури та вологості DHT22 (AM2302):

а) стандартний сенсор; б) сенсор у великому кейсі з проводами [1-2]

Взаємодія з сенсором DHT22 виконується по протоколу I<sup>2</sup>C (Inter-Integrated Circuit, послідовна шина даних для зв'язку інтегральних схем). Діапазон вимірювання відносної вологості: від % до 100% з похибкою до 2%. Діапазон вимірювання температури сенсора DHT22 від -40 °C до 80 °C з похибкою до 0.5 °C. Таким чином, сенсор DHT22 придатний для практичного вимірювання параметрів мікроклімату.

### **1.3. Аналіз комп'ютерних систем-аналогів для зчитування сигналів із сенсорів**

Зчитування сигналів з сенсорів виконується за допомогою комп'ютерів, мікрокомп'ютерів та контролерів. Використання мікрокомп'ютерів та контролерів доцільне для побудови компактних систем низької та середньої складності.

Зчитування сигналів з сенсорів часто виконується за допомогою Arduino (рис. 1.3). Пристрій Arduino складається з мікроконтролера Atmel AVR, а також додаткових елементів для програмування та інтеграції з іншими пристроями.



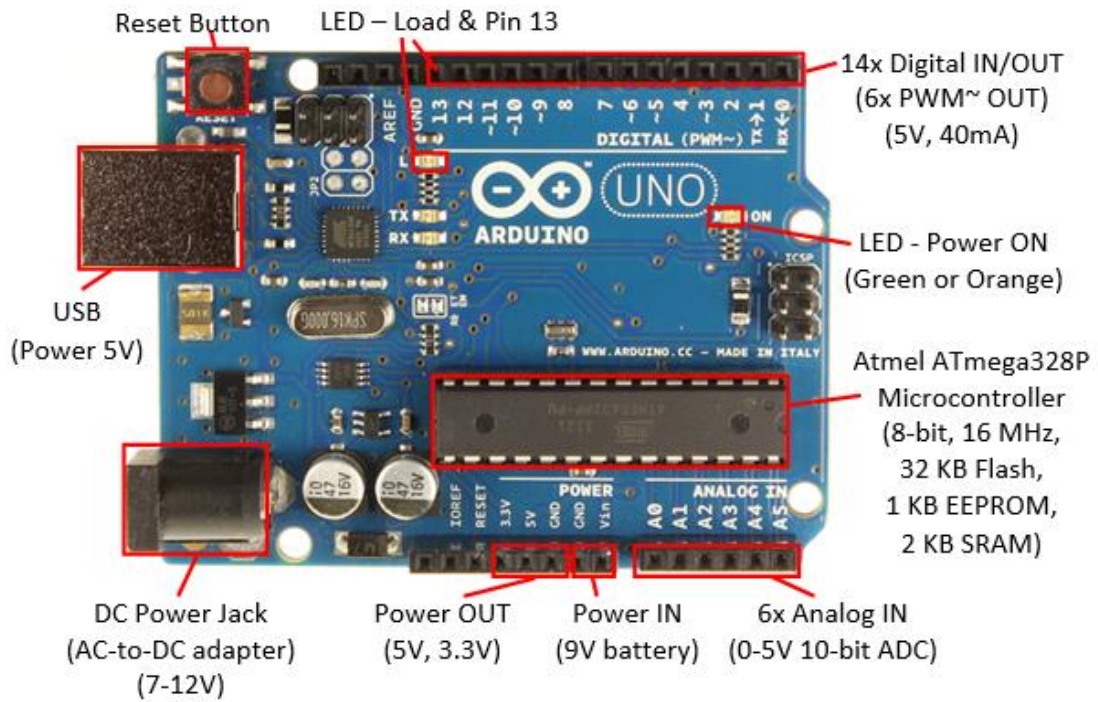
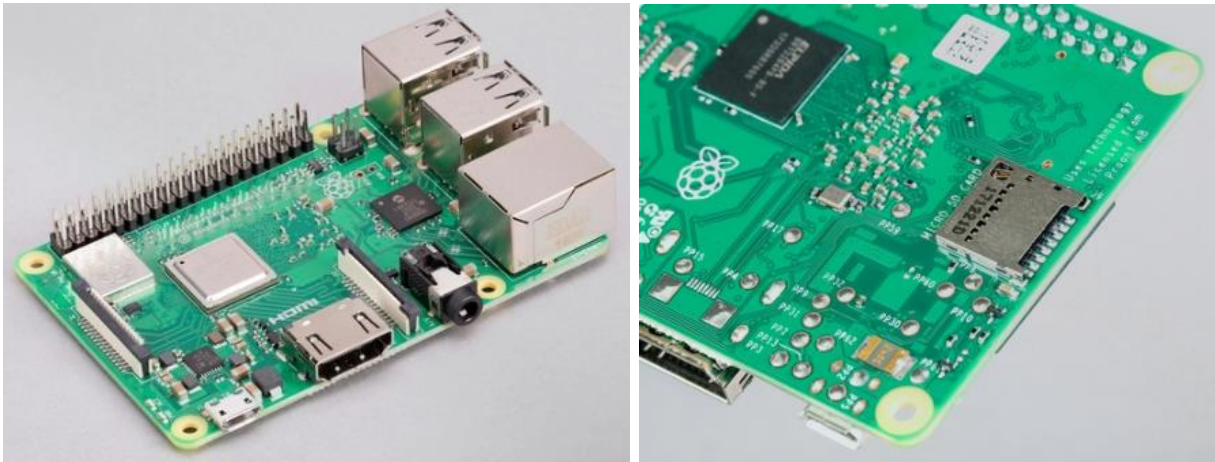


Рисунок 1.3 – Зовнішній вигляд та сигнали пристрою Arduino Uno [9]

Напруга живлення пристрою +5В. Тактова частота контролера складає 16 або 8 МГц. У мікроконтролер записаний завантажувач, тому для Arduino зовнішній програматор не потрібен. На концептуальному рівні усі пристрою Arduino програмуються через Arduino (RS-232), але нові пристрої програмуються через шину USB завдяки мікросхемі конвертера USB-to-Serial FTDI FT232R. В пристрої Arduino Uno як конвертер використовується контролер Atmega8. Пристрої Arduino дозволяють використовувати значну кількість входів та виходів (пінів) для під'єднання зовнішніх пристроїв. Наприклад, у пристрої Decimila є 14 цифрових входів/виходів (напрямо передачі сигналів встановлюється програмно), 6 із яких можуть генерувати сигнал з широтно-імпульсною модуляцією, а також міститься 6 аналогових входів.

Зчитування сигналів із сенсорів також виконується за допомогою мікрокомп'ютера Raspberry Pi3 (рис. 1.4), який дозволяє зчитувати сигнали різних типів сенсорів (в тому числі й DHT22) і проводити їх обробку в режимі реального часу.



а)

б)

Рисунок 1.4 – Одноплатний комп'ютер Raspberry Pi 3 Model B+ :  
а) вигляд зі сторони процесора; б) зі сторони карти мікро SD [1-2]

Мікрокомп'ютер Raspberry Pi3 керується операційною системою Raspbian [1-2], яка встановлюється на мікро SD карту. Основними параметрами мікрокомп'ютера Raspberry Pi 3 Model B+ є наступні:

- Процесор Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-біт з тактовою частотою 1.4 ГГц.
- Оперативна пам'ять об'ємом 1ГБ, тип LPDDR2 SDRAM.
- WiFi IEEE 802.11.b/g/n/, Bluetooth 4.2.
- Порт Gigabit Ethernet.
- 40-піновий конектор GPIO (General-Purpose input/output, інтерфейс введення виведення загального призначення, через який можливе під'єднання сенсорів).
- Конектор HDMI для приєднання монітору.
- 4 порта USB 2.0.
- Порт CSI (Camera Serial Interface) для під'єднання відеокамери.
- Порт DSI (Display Serial Interface) для під'єднання дисплею.
- Порт аудіовиходу.
- Порт мікро SD.

Живлення Raspberry Pi3B+ повинно виконуватися зарядним пристроєм з потужністю з напругою живлення 5В і вихідним струмом не менше 3А.

Перевагою Raspberry Pi3 є компактність і досить висока швидкодія, тому в даній роботі для зчитування сигналів з сенсорів використано мікрокомп'ютер Raspberry Pi3.

Існуючі комп'ютерні системи на базі Raspberry Pi 3 дозволяються зчитувати та зберігати сигнали з різних видів сенсорів температури та вологості. Проте, такі системи звичайно не використовують глибший аналіз сигналів, наприклад, шляхом кластеризації. Такий аналіз може бути корисний при дослідженні мікроклімату приміщень, тому в даній роботі запропоновано розробити комп'ютерну систему на базі Raspberry Pi 3, яка буде не тільки зчитувати і зберігати сигнали температури і вологості, але й проводити їх кластеризацію.

#### **1.4. Постановка завдання, опис вимог до системи на основі аналізу аналогів**

На основі аналізу систем-аналогів сформульовано такі вимоги до системи аналізу значень температури і вологості:

1. Значення сигналів зчитувати з інтегрованого сенсора температури та вологості DHT22.
2. Комп'ютерну систему для зчитування, збереження та аналізу сигналів температури і вологості розробити на основі мікрокомп'ютера Raspberry Pi3, який є компактным і має достатню швидкодію.
3. Для дослідження мікроклімату приміщень проводити аналіз сигналів температури і вологості методами кластеризації.
4. Програму для обробки сигналів розробити на мові Python.

## 2. ОГЛЯД ПІДХОДІВ ДО ВИРІШЕННЯ ЗАВДАННЯ

### 2.1. Загальна будова комп'ютерної системи для аналізу сигналів температури і вологості

Апаратна частина комп'ютерної системи аналізу значень температури та вологості, зчитаних з сенсорів, складається з одноплатного комп'ютера Raspberry Pi3B+ [3-4] та сенсора температури DHT22 (рис. 2.1).

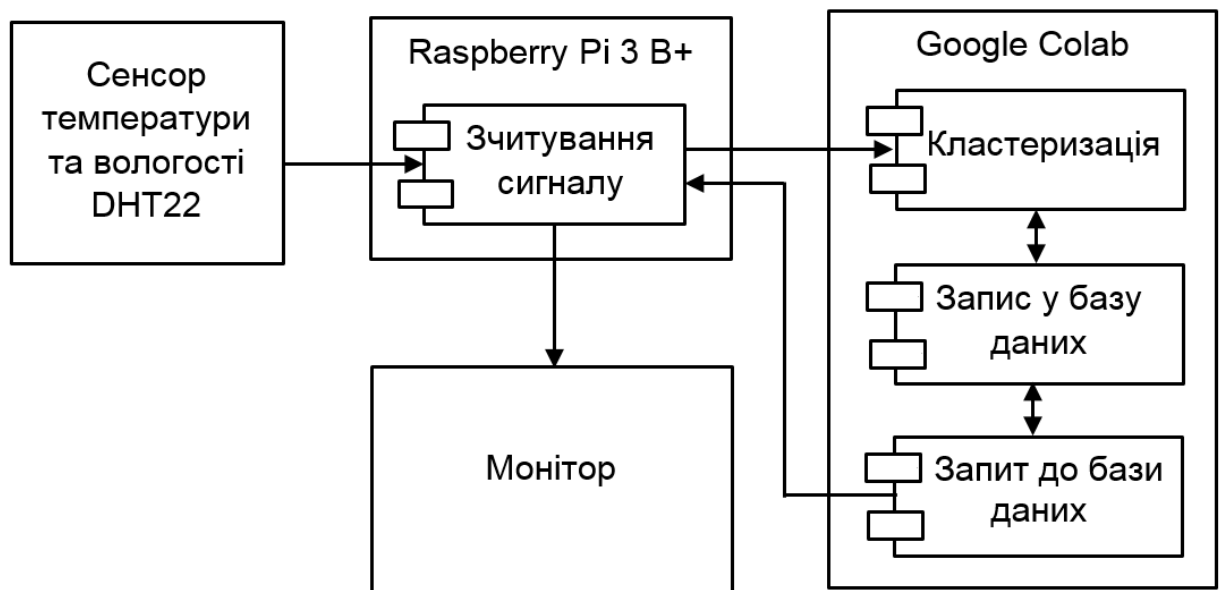


Рисунок 2.1 – Фізична модель системи аналізу значень температури та вологості (UML діаграма розміщення / deployment diagram)

Керування одноплатним комп'ютером Raspberry Pi 3 Model B+ виконується операційною системою Raspbian, яка завантажується з мікро SD картки (рис. 2.1). Сигнали з сенсора температури та вологості DHT22 зчитуються через піни GPIO. Через USB конектори приєднується клавіатура, маніпулятор «миша», шлеш-накопичувач на інші пристрої. Конектор HDMI призначений для виведення зображення на монітор.

Згідно з фізичною моделлю системи аналізу значень температури та вологості сигнали з сенсора DHT22 зчитуються в Raspberry Pi 3 Model B,

виводяться на монітор і зберігаються у файли за допомогою програми «read\_sensor\_TH» (на мові Python).

Через мережеве з'єднання (WiFi) зчитані сигнали передаються програмі «b\_Cluster\_TH» (на мові Python), яка виконується на хмарній платформі Google Colab. Програма «b\_Cluster\_TH» забезпечує кластеризацію сигналів у просторі ознак температура-вологість, збереження отриманих значень в базу даних SQLite і повертає результати запитів до бази даних.

## 2.2. Основи кластеризації даних

При виміряних сигналів температури та вологості доцільно проводити кластеризацію у просторі ознак саме температури і вологості. Задача кластеризації (поряд із задачею класифікації) є важливою для практики задачею інтелектуального аналізу даних [5-7]. Поняття «кластер» (cluster) перекладається як «скупчення», «гроно». Можна розглядати кластеризацію як логічне продовження класифікації. Особливістю кластеризації є те, що класи об'єктів не визначені перед початком аналізу. У результаті кластеризації всі об'єкти розділяються за певними ознаками на кластери (групи). Якщо досліджувані об'єкти представити як точки в просторі ознак, то задача кластеризації буде зводитися до визначення компактно розміщених точок («згущень точок» або кластерів).

Кластери можна охарактеризувати як групу (множину) об'єктів, які мають деякі спільні характеристики (властивості). Кластери мають такі дві характерні особливості:

- внутрішня однорідність кластеру;
- зовнішня ізольованість (відносно інших кластерів).

Сучасні методи кластеризації засновані на тому, що для створення ефективних класифікацій об'єктів потрібно розглядати різноманітні характеристики досліджуваних об'єктів, оцінювати міри схожості між цими об'єктами і розміщувати схожі об'єкти в одному кластері.

На даний час запропоновано значну кількість алгоритмів кластеризації: методи  $k$ -середніх, ієрархічні методи. Алгоритми кластеризації постійно удосконалюються за точністю і швидкодією.

Існує три причини широкого застосування кластерного аналізу. По-перше, це доступність потужної комп'ютерної техніки (у тому числі й хмарні сервіси), яка дозволяє проводити кластерний аналіз для великих об'ємів даних за допустимий час. По-друге, в сучасній науці та техніці посилюється спеціалізація знань, тому існує потреба у впорядкуванні отриманих даних. Третьою причиною є збільшення кількості параметрів, які використовуються при аналізі досліджуваних об'єктів.

Розглянемо формальну постановку задачі кластеризації. Початковими даними є навчальна вибірка об'єктів  $T = (t_1, \dots, t_Q)$ , а також функція відстані між об'єктами  $\rho(t_i, t_k)$ , де  $Q$  – кількість об'єктів. Потрібно розділити об'єкти вибірки на кілька підмножин (кластерів) так, щоб кожен кластер складався з об'єктів, які є близькими по відстані (метриці)  $\rho$ . При цьому об'єкти різних кластерів повинні суттєво відрізнятися. У результаті кластеризації кожному об'єкту  $t_i$  ставиться у відповідність мітка (номер) кластера  $y_j$  ( $j = 1, \dots, k$ , де  $k$  – кількість кластерів).

Кількість кластерів  $k$  у деяких випадках відома ще до кластеризації, проте частіше потрібно визначити оптимальне число кластерів з точки зору певного критерію якості кластеризації.

На даний час потрібно обробляти великі об'єми даних (наприклад, отриманих експериментально за допомогою сенсорів температури і тиску), тому існують такі вимоги для алгоритмів кластеризації:

- мінімально можлива кількість ітерацій для забезпечення високої швидкодії;
- використання обмеженого об'єму оперативної пам'яті комп'ютера;
- можливість переривання роботи алгоритму із збереженням проміжних результатів та подальшим продовженням кластеризації.

Вибір відстані  $\rho(t_i, t_k)$  між досліджуваними об'єктами  $t_i$  та  $t_k$  є важливим етапом аналізу, оскільки від нього суттєво залежить остаточний варіант поділу об'єктів на кластери при певному алгоритмі кластеризації. Кожен досліджуваний об'єкт  $t_i$  описується  $D$  ознаками, як позначаються  $X_{i,v}$ , де  $v = 1, \dots, D$ . Якщо об'єкти описуються двома ознаками  $X_{i,1}$  та  $X_{i,2}$  ( $D = 2$ ), то такі об'єкти можна зобразити як точки на площині з просторовими декартовими координатами  $x_1$  та  $x_2$  (рис. 2.2). Наприклад, при аналізі сигналів з сенсорів температури та вологості як координату  $x_1$  об'єкта (виміру) можна розглядати значення температури, а як координату  $x_2$  – значення вологості.

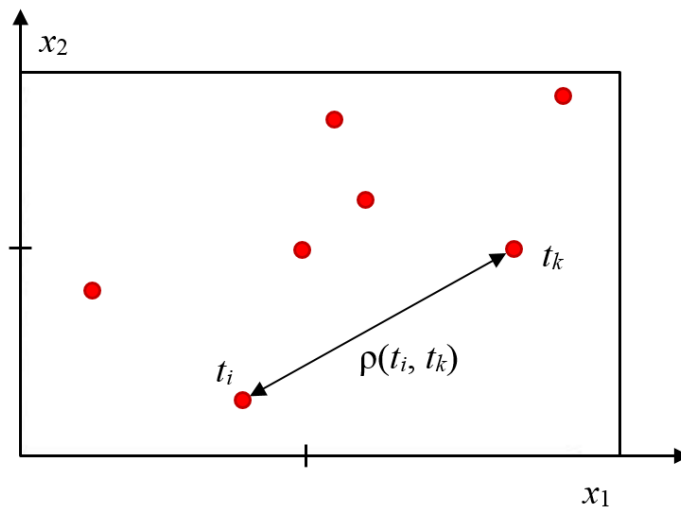


Рисунок 2.2 – Обчислення відстані (метрики) між об'єктами  $t_i$  та  $t_k$

На даний час використовується декілька методів визначення функції відстані  $\rho$ :

1. Відстань Евкліда, яка є геометричною відстанню в багатовимірному просторі й обчислюється за формулою:

$$\rho(t_i, t_k) = \sqrt{\sum_{v=1}^D (X_{i,v} - X_{k,v})^2}. \quad (2.1)$$

де  $X_{i,v}$  – ознаки об'єктів.

Наприклад, при використанні як ознак об'єктів значень температури та вологості формулу (2.1) можна записати у вигляді:

$$\rho(t_i, t_k) = \sqrt{(T_i - T_k)^2 + (H_i - H_k)^2}, \quad (2.2)$$

де  $T_i$  – температура об'єкту  $i$  (виміру з номером  $i$ );

$T_k$  – температура об'єкту  $k$  (виміру з номером  $i$ );

$H_i$  – вологість об'єкту  $i$  (виміру з номером  $i$ );

$H_k$  – вологість об'єкту  $k$  (виміру з номером  $k$ ).

Відстань Евкліда як критерій кластеризації має певні переваги. Наприклад, відстань між двома об'єктами не змінюється при введенні у вибірку нового об'єкту (виміру), який може виявитися похибкою. Тому в даній роботі запропоновано визначати відстань між об'єктами як відстань Евкліда.

2. Відстань міських кварталів або манхеттенська відстань, яка обчислюється через різницю координат об'єктів. У більшості випадків ця міра відстані дає ті ж результати кластеризації, що й відстань Евкліда. Проте, для цієї міри вплив окремих великих різниць параметрів (викидів) зменшується, оскільки вони не підносяться до квадрату. Манхеттенська відстань обчислюється за формулою

$$\rho(t_i, t_k) = \sqrt{\sum_{v=1}^D |X_{i,v} - X_{k,v}|}. \quad (2.3)$$

3. Відстань Чебишева може бути корисною, якщо потрібно розділити два об'єкти на рівні кластери за різницею по одній координаті (за одним параметром). Відстань Чебишева обчислюється за формулою

$$\rho(t_i, t_k) = \max(X_{i,v} - X_{k,v}), \quad (2.4)$$

де  $v = 1, \dots, D$ .

Поширеним методом кластеризації є метод  $k$ -середніх, який програмно реалізовано, наприклад, мовою Python [10] в бібліотеці `sklearn`.



### **2.3. Функції комп'ютерної системи**

До основних функцій комп'ютерної системи для аналізу значень температури і вологості належать:

1. Зчитування початкових значень температури і вологості з сенсора.
2. Візуалізація отриманих значень і збереження їх у файл.
3. Кластеризація значень температури і вологості, збереження їх у базі даних.

### 3. ПРОЕКТУВАННЯ КОМП'ЮТЕРНОЇ СИСТЕМИ

#### 3.1. Кластеризація методом k-середніх

Метод k-середніх є поширеним методом кластеризації, який програмно реалізовано, наприклад, мовою Python [10] в бібліотеці sklearn. Така кластеризація дозволяє об'єднувати значення параметрів температури і вологості у групи подібних, що спрощує їх подальших аналіз. При аналізі великої кількості вимірів можливо виявити кластери з подібним мікрокліматом (близькими значеннями температури і вологості). Модель кластерного аналізу вимірів передбачає, що кожен з  $Q$  вимірів описується  $D = 2$  параметрами: температурою – параметр  $P(i,0)$  та вологістю – параметр  $P(i,1)$ , де  $i = 1, \dots, Q$ .

Для знаходження відстані між вимірами  $t_i$  та  $t_j$  використано відстань Евкліда  $\rho(t_i, t_j)$  (2.1).

Для кількісної оцінки якості кластеризації застосовується внутрішньокластерна відстань (3.1) та міжкластерна відстань (3.2).

Середня внутрішньокластерна відстань має бути якомога меншою:

$$F_0 = \frac{\sum_{i=1}^Q \sum_{k=1}^Q [y_i = y_k] \rho(t_i, t_k)}{\sum_{i=1}^Q \sum_{k=1}^Q [y_i = y_k]} \rightarrow \min, \quad (3.1)$$

де  $y_i$  – номер кластера, до якого належить об'єкт з номером  $i$ ;

$y_k$  – номер кластера, до якого належить об'єкт з номером  $k$ ;

$Q$  – кількість об'єктів (вимірів).

Середня міжкластерна відстань має бути якомога більшою:

$$F_1 = \frac{\sum_{i=1}^Q \sum_{k=1}^Q [y_i \neq y_k] \rho(t_i, t_k)}{\sum_{i=1}^Q \sum_{k=1}^Q [y_i \neq y_k]} \rightarrow \max. \quad (3.2)$$

На практиці обчислюють відношення пари критеріїв для врахування як міжкластерних, так і внутрішньокластерних відстаней за формулою:

$$F_0 / F_1 \rightarrow \min . \quad (3.3)$$

Максимізація відношення (3.3) реалізована у функції KMeans бібліотеки sklearn.cluster при кластеризації даних.

Оцінка якості кластеризації може бути проведена на основі таких процедур:

- ручна перевірка результатів кластеризації;
- встановлення контрольних точок і перевірка на отриманих кластерах;
- визначення стабільності кластеризації шляхом додавання нових об'єктів;
- порівняння результатів, отриманих різними методами кластеризації.

В методі k-середніх кількість кластерів k потрібно вказати перед кластеризацією.

### 3.2. Апаратно-програмна реалізація комп'ютерної системи

Апаратна частина комп'ютерної системи аналізу значень температури та вологості, зчитаних з сенсорів, складається з одноплатного комп'ютера Raspberry Pi 3B+ та сенсора температури DHT22 (рис. 3.3).

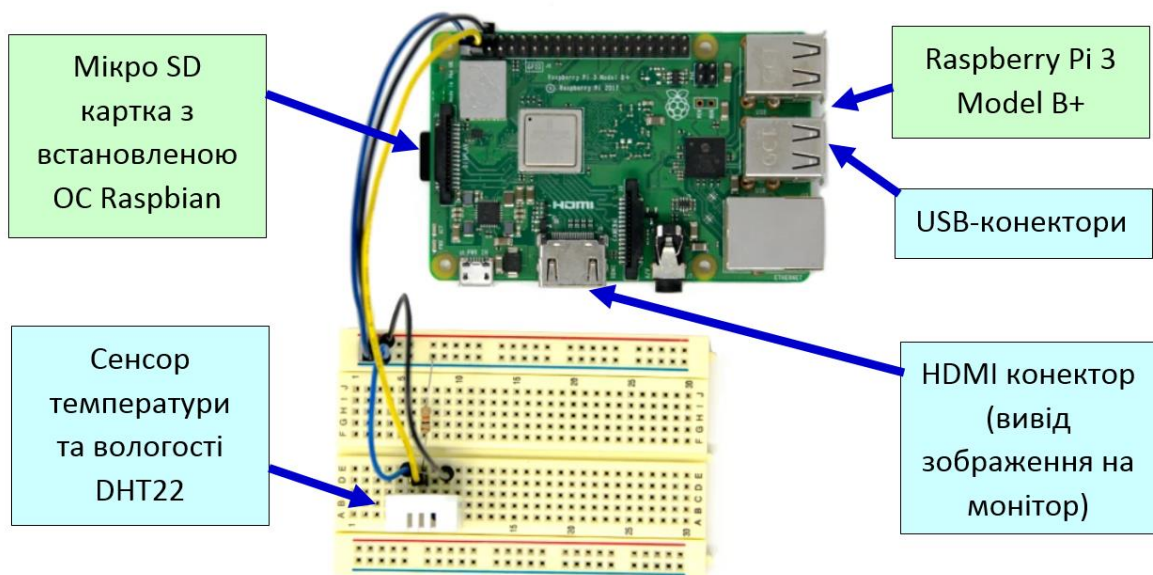


Рисунок 3.3 – Під'єднання сенсора температури та вологості DHT22 до Raspberry Pi 3 Model B+

Програма «read\_sensor\_TH» (на мові Python) призначена для зчитування значень температури та вологості з сенсора DHT22 за допомогою мікрокомп'ютера Raspberry Pi. В програмі імпортується бібліотека Adafruit\_DHT для роботи з сенсором DHT22, зчитування сигналу виконується через пін GPIO4 (DHT\_PIN = 4).

Згідно з алгоритмом (рис. 3.4) встановлюється початковий час зчитування  $t_m$  (с), інтервал дискретизації /час між зчитуваннями/  $\Delta t$  (с), максимальна допустима температура  $Limit\_TM$ ; кількість значень  $QP$ , які потрібно зчитати. Далі в циклі зчитуються значення температури (записуються в масив  $mTR$ ) та значення вологості (записуються в масив  $mTH$ ). Час кожного зчитування записується в масив  $mtcR$ . Значення вологості ( $humidity$ ) та температури ( $temperature$ ) зчитуються з сенсора командою

```
humidity, temperature = Adafruit_DHT.read_retry(DHT_SENSOR, DHT_PIN)
```

Якщо зчитане значення температури  $mTR[i]$  перевищує допустиме  $Limit\_TM$ , то в масив допустимих значень  $mTR\_Lim$  записується значення 1 (а інакше значення 0) і виводиться відповідне повідомлення.

Мінімальні та максимальні значення температури кластерів зберігаються в масивах  $TR\_min$  та  $TR\_max$  відповідно. Мінімальні та максимальні значення вологості кластерів зберігаються в масивах  $TH\_min$  та  $TH\_max$  відповідно.

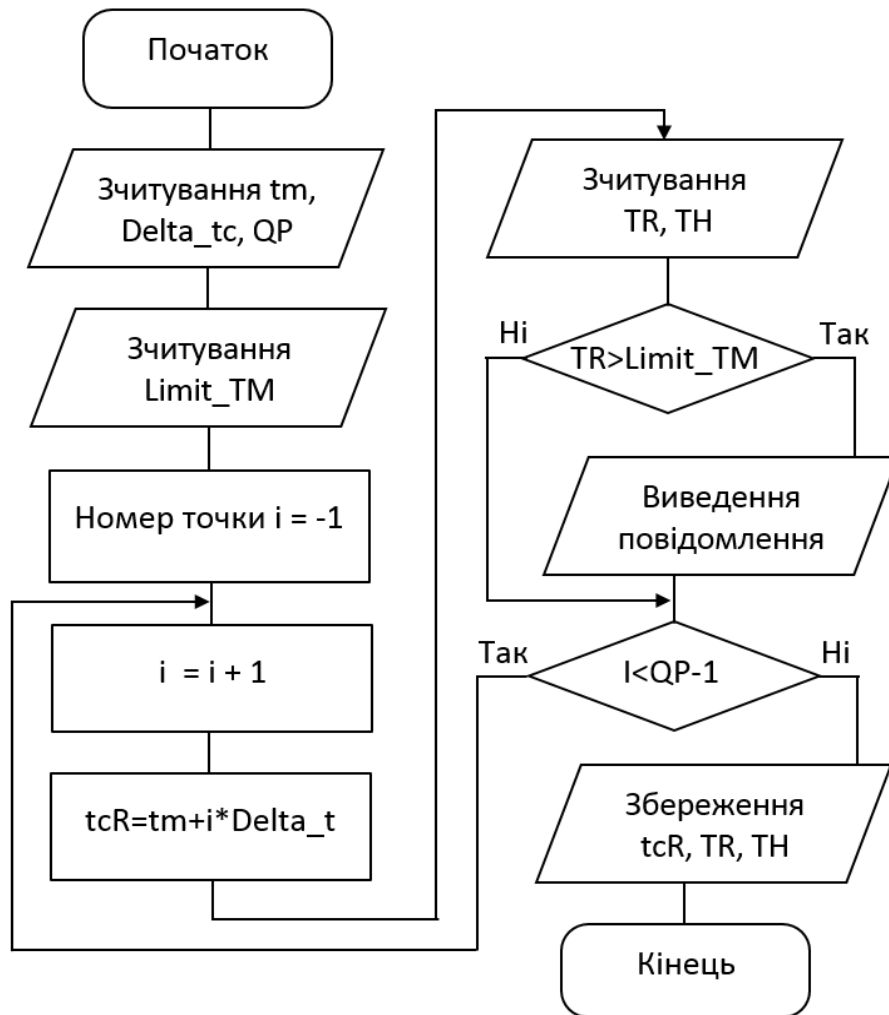


Рисунок 3.4 – Схема алгоритму зчитування сигналів температури і вологості з сенсора

Програма «read\_sensor\_TH виконується в середовищі Thonny (рис. 3.5).

```

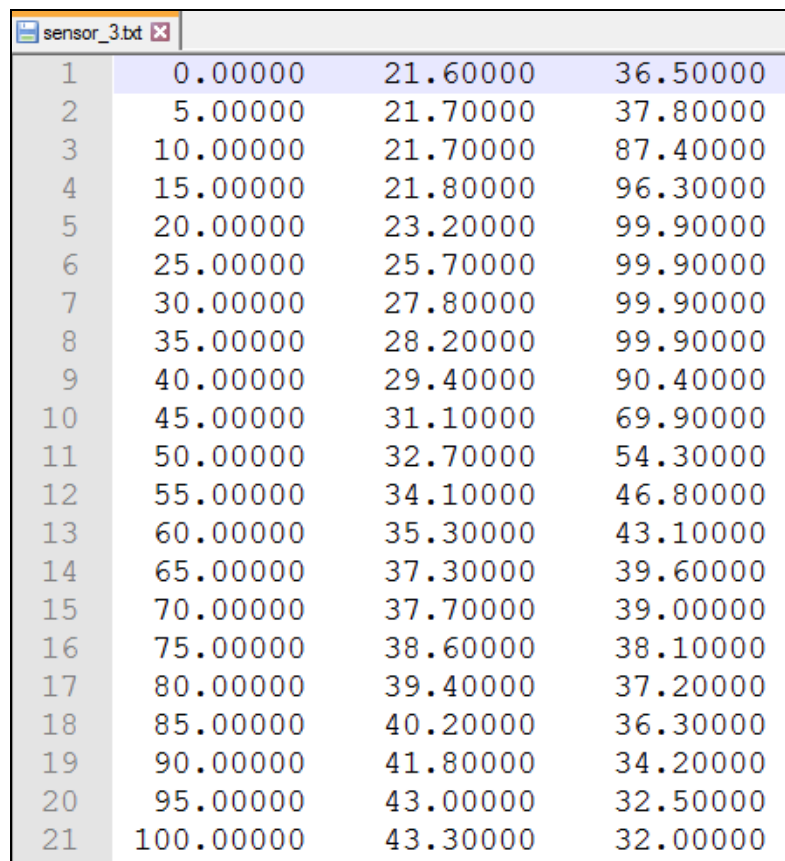
read_sensor_TH.py
1
2
3
4
5
6 import numpy as np # імпорт бібліотеки для математичних обчислень
7 import time
8 import Adafruit_DHT
9 DHT_SENSOR = Adafruit_DHT.DHT22
10 DHT_PIN = 4 # номер піна GPIO4
11
12 Time_Min=0 # початковий час зчитування, с
13 #Delta_tc=900 # 900с, 15 min
14 Delta_tc=5.0 # 2 с
15 #Delta_t=Delta_tc/3600 # інтервал дискретизації (час між зчитуваннями, години)
16 Delta_t=Delta_tc # інтервал дискретизації (час між зчитуваннями, с)
17 Limit_TM=50 # максимальна допустима температура
  
```

Рисунок 3.5 – Виконання програми «read\_sensor\_TH в середовищі Thonny

Якщо зчитане значення температури  $mTR[i]$  та вологості  $mTH[i]$  знаходяться в допустимих межах для кластера з номером  $nCL$ , то в масив  $mCL$  записується номер кластера і виводиться повідомлення. Програмно така перевірка виконується кодом:

```
if (mTR[i]>TR_min[nCL]) AND (mTR[i]<TR_max[nCL]):
    if (mTH[i]>TH_min[nCL]) AND (mTH[i]<TH_max[nCL]):
        mCL[i]=nCL
        print('Кластер nCL=',nCL)
```

Отримані масиви значень ( $mtcR$ ,  $mTR$ ,  $mTH$ ) записуються в масив  $mtc\_TRH$ , який зберігається в текстовий файл (рис. 3.6) командою `np.savetxt("file_name.txt",mtc_TRH, fmt='% 10.5f',delimiter='\t')`



Row	Column 1	Column 2	Column 3
1	0.00000	21.60000	36.50000
2	5.00000	21.70000	37.80000
3	10.00000	21.70000	87.40000
4	15.00000	21.80000	96.30000
5	20.00000	23.20000	99.90000
6	25.00000	25.70000	99.90000
7	30.00000	27.80000	99.90000
8	35.00000	28.20000	99.90000
9	40.00000	29.40000	90.40000
10	45.00000	31.10000	69.90000
11	50.00000	32.70000	54.30000
12	55.00000	34.10000	46.80000
13	60.00000	35.30000	43.10000
14	65.00000	37.30000	39.60000
15	70.00000	37.70000	39.00000
16	75.00000	38.60000	38.10000
17	80.00000	39.40000	37.20000
18	85.00000	40.20000	36.30000
19	90.00000	41.80000	34.20000
20	95.00000	43.00000	32.50000
21	100.00000	43.30000	32.00000

Рисунок 3.6 – Текстовий файл зі зчитаними значеннями температури та вологості

Програма «b\_Cluster\_TH» (на мові Python) для кластеризації сигналів з сенсорів виконується на хмарній платформі Google Colab [12]. Згідно з алгоритмом (рис. 3.7) спочатку зчитуються значення часу  $t_{cR}$ , температури  $TR$  (рис. 3.8а) і вологості  $TH$  (рис. 3.8б). Кластеризація виконується методом  $k$ -середніх ( $kmeans$ ) [5-7] за ознаками температура  $TR$  і вологість  $TH$  (рис. 3.9). Після встановлення кількості кластерів  $k$  кластеризація виконується функцією  $kmeans.fit$ . У результаті кластеризації для кожного зчитаного значення обчислюється його номер кластеру  $N_c$ . Для кластерів обчислюються координати їх центрів  $C_s$  за температурою та вологістю, мінімальне значення за температурою  $TR_{min}$ , максимальне значення за температурою  $TR_{max}$ , мінімальне значення за вологістю  $TH_{min}$ , максимальне значення за вологістю  $TH_{max}$ . Отримані параметри кластерів зберігаються у файл та в базу даних SQLite.

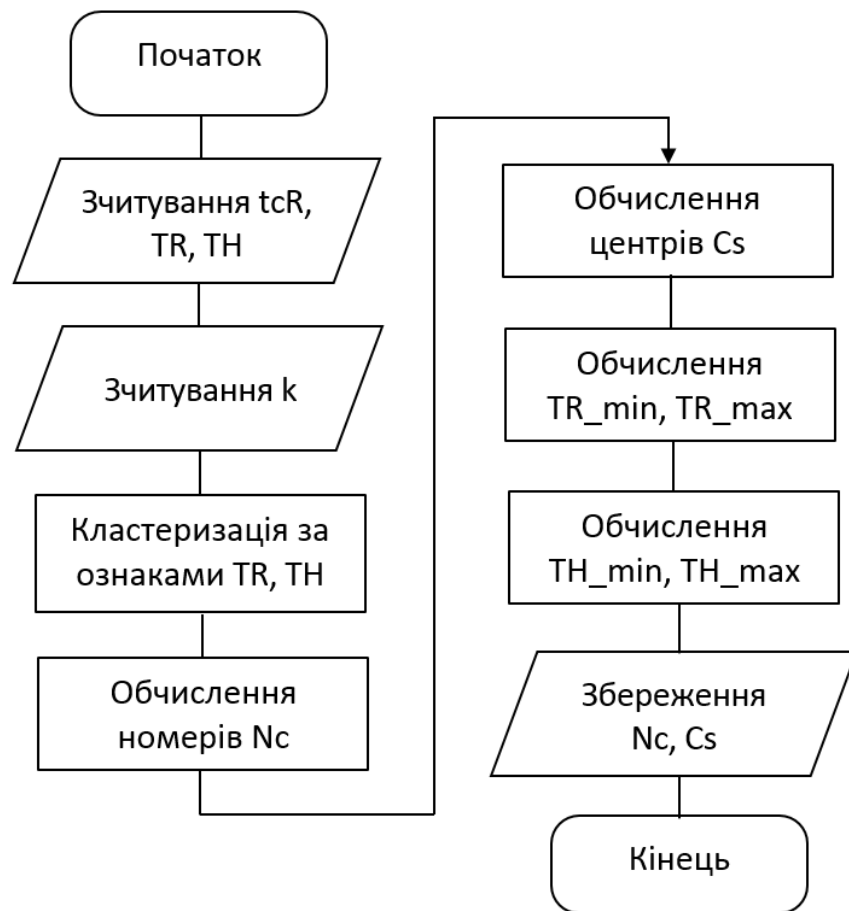


Рисунок 3.7 – Схема алгоритму програми для кластеризації сигналів

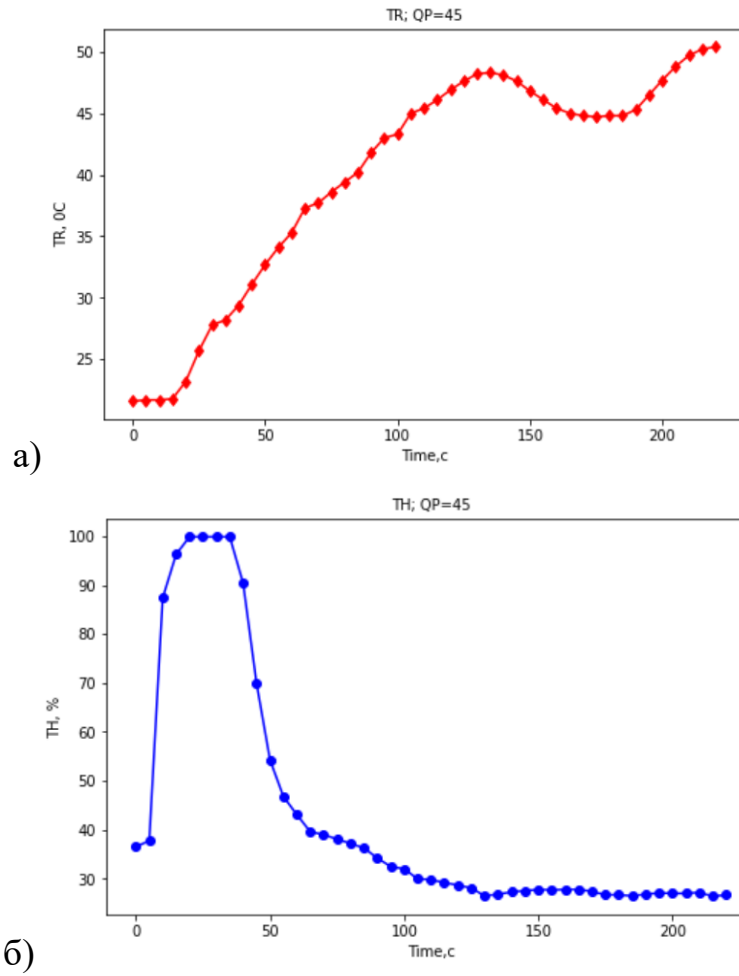


Рисунок 3.8 – Приклад графіків залежності значень температури TR та вологості TH від часу

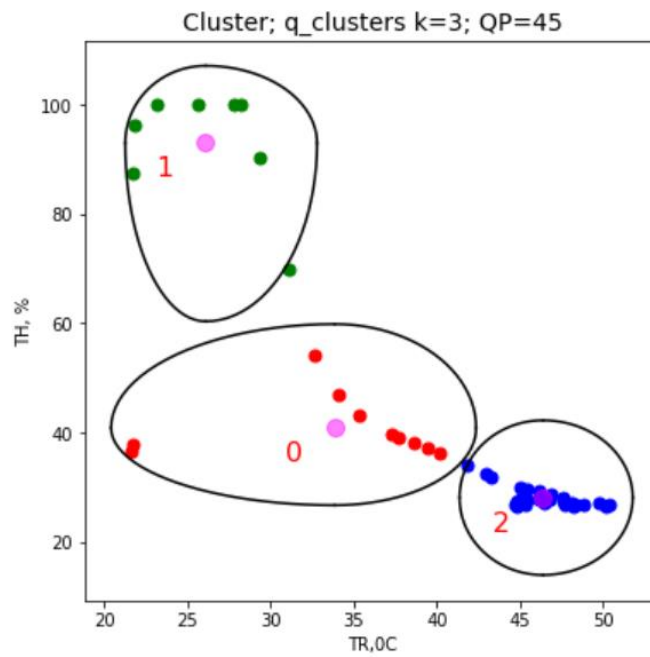
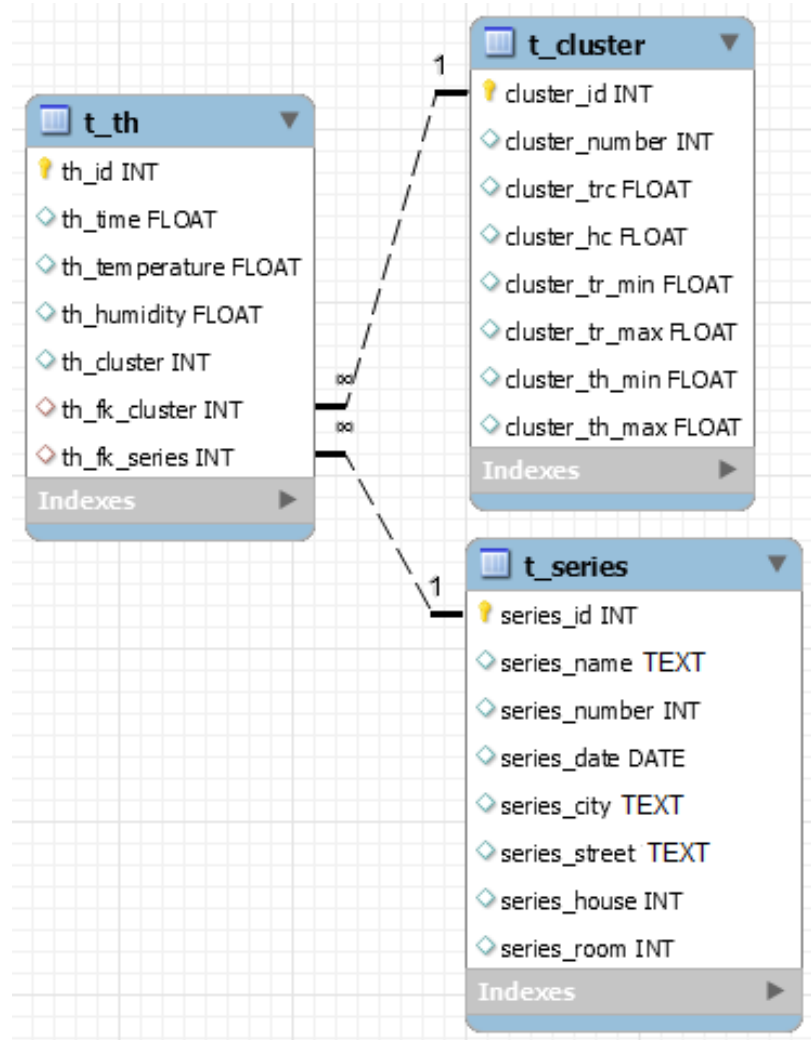


Рисунок 3.9 – Кластеризація значень вимірів (рис. 3.8) у просторі ознак температура (TR) та вологість (TH)



Для збереження отриманих значень використовується база даних «db\_sensor\_TH» (рис. 3.10) системи керування базами даних SQLite [12], операції з якою виконуються на хмарному сервісі Google Colab.



б)

Рисунок 3.10 – ER діаграма (діаграма сутність-зв’язок) фізичної моделі бази даних «db\_sensor\_TH» (MySQL Workbench)

Таким чином, розроблена програма як складова частина системи аналізу значень температури і вологості дозволяє користувачу виконувати ряд таких завдань (рис. 3.11).



Рисунок 3.11 – Діаграма прецедентів системи аналізу значень температури і вологості

Користувач може встановити параметри зчитування сигналу (наприклад, час дискретизації), параметри кластеризації (наприклад, час кластрів). Кластеризація виконується методом k-середніх, результати кластеризації візуалізуються і зберігаються в базі даних.

## ВИСНОВКИ

У результаті виконання переддипломної практики отримано такі результати:

1. Проведено опис сфери використання комп'ютерних систем для аналізу значень температури і вологості, виконано аналіз системи-аналогів.
2. Розроблено комп'ютерну систему аналізу значень температури та вологості, яка складається з одноплатного комп'ютера Raspberry Pi3B+, сенсора температури DHT22 та керуючих програм на мові Python.
3. Розроблено програму «read\_sensor\_TH» на мові Python для зчитування значень температури та вологості з сенсора DHT22.
4. Розроблено програму «b\_Cluster\_TH» на мові Python, яка виконується для кластеризації значень температури та вологості методом k-середніх.
5. Створено базу даних SQLite для збереження результатів кластеризації.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Digital-output relative humidity & temperature sensor/module DHT22 (DHT22 also named as AM2302). Capacitive-type humidity and temperature. URL: <https://datasheetspdf.com/mobile/792211/Aosong/DHT22/1>.
2. DHT22 Raspberry Pi Humidity Temperature Sensor Tutorial. URL: <https://www.youtube.com/watch?v=IHTnU1T8ETk>.
3. Raspberry Pi Compute Module 3+. URL: <https://datasheets.raspberrypi.com/cm/cm3-plus-datasheet.pdf>.
4. Raspberry Pi 3 Model B+. URL: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>.
5. Черняк О.І., Захарченко П.В. Інтелектуальний аналіз даних: Підручник. Київський національний університет ім. Т. Шевченка. К. : Знання, 2014. 599 с.
6. Глибовець М.М., Олецький О.В. Штучний інтелект. К. : КМ Академія, 2002. 336 с.
7. Штовба С.Д., Мазуренко В.В. Інтелектуальні технології ідентифікації залежностей. Лабораторний практикум: електронний навчальний посібник. Вінниця : ВНТУ, 2014. 113 с.
8. Вимірювальні перетворювачі (сенсори): підручник / за ред. Є.С. Поліщука та В. М. Ванька. Львів : Видавництво Львівської політехніки, 2015. 584 с.
9. Arduino. URL: <https://www.arduino.cc>.
10. Яковенко А. В. Основи програмування. Python. Частина 1: підручник. Київ : КПІ ім. Ігоря Сікорського, 2018. 195 с. URL: <https://ela.kpi.ua/handle/123456789/25111>.
11. Thonny. Python IDE. URL: <https://thonny.org>.
12. Google Colab. URL: <https://colab.research.google.com>.