



# System do Monitorowania Jakości Powietrza i Mikroklimatu w pomieszczeniach

Podsumowanie prac – Semestr I: Innowacyjne podejście do kontroli środowiska wewnętrznego.

1

## Autonomiczny Węzeł Pomiarowy

Stworzenie niezależnej jednostki do precyzyjnych pomiarów.

2

## Integracja Wielu Protokołów

Obsługa komunikacji I2C i UART dla różnorodnych sensorów.

3

## Wizualizacja Danych w Czasie Rzeczywistym

Intuicyjny dashboard do monitorowania bieżących parametrów.

# Architektura Systemu: Od Sensora do Ekranu

Kompleksowe rozwiązanie do monitorowania jakości powietrza, oparte na modułowej budowie.



## Hardware

Sensoryka i połączenia fizyczne

## Backend

Kolektor danych i przetwarzanie

## Frontend

Wizualizacja i interfejs użytkownika

1

### HARDWARE

Sensoryka i Połączenia Fizyczne

Odpowiedzialny za akwizycję danych z otoczenia.

2

### BACKEND

Kolektor Danych i Baza

Zarządzanie danymi: Python (kolektor) + SQLite (baza danych).

3

### FRONTEND

Dashboard Webowy

Interfejs użytkownika: Flask + JavaScript do wizualizacji danych.

# Komponenty Hardware: Pomiary Gazowe

Wybór wysokiej klasy czujników zapewnia dokładność i niezawodność pomiarów.

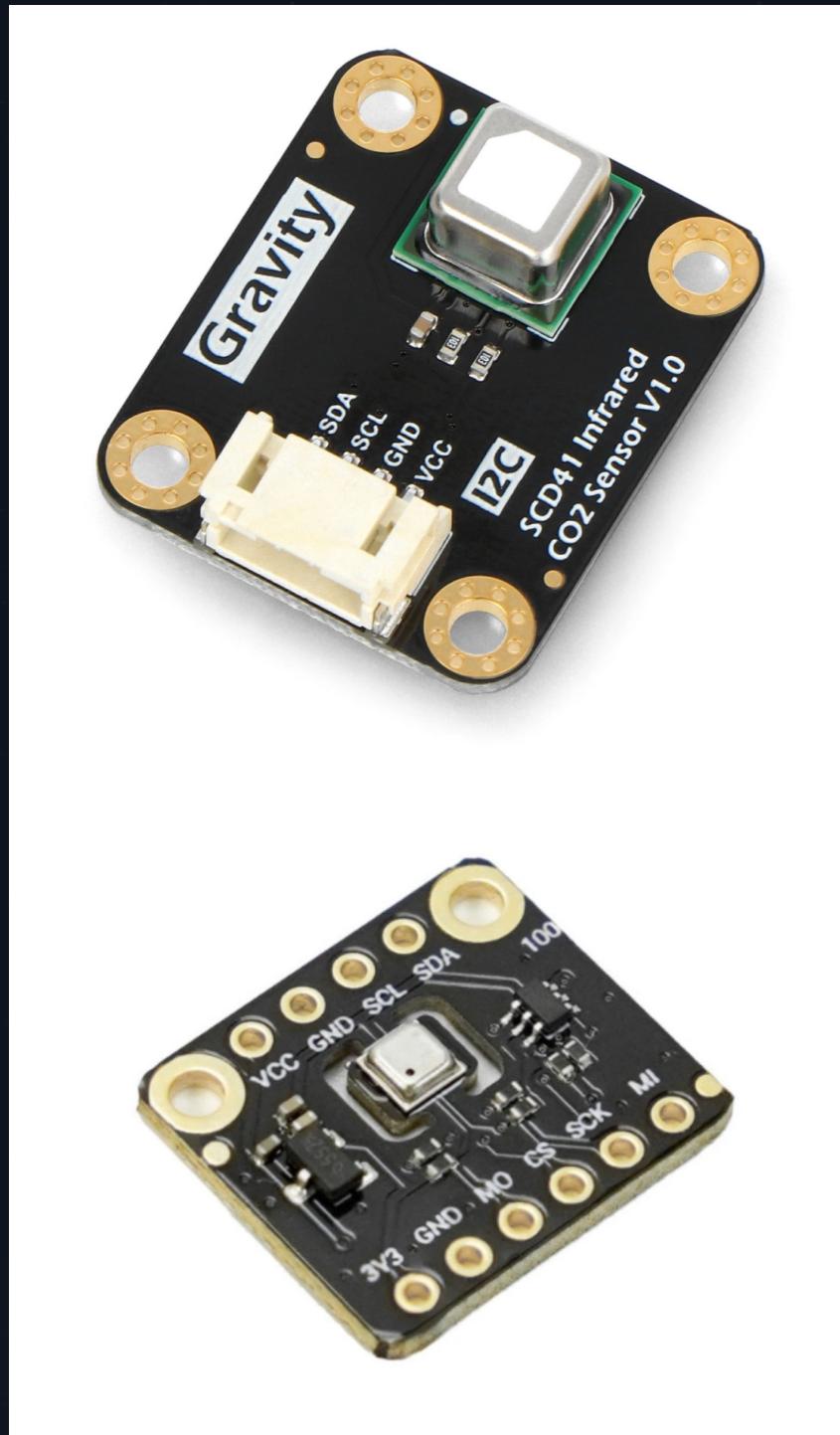
## Czujnik SCD41 (CO2)

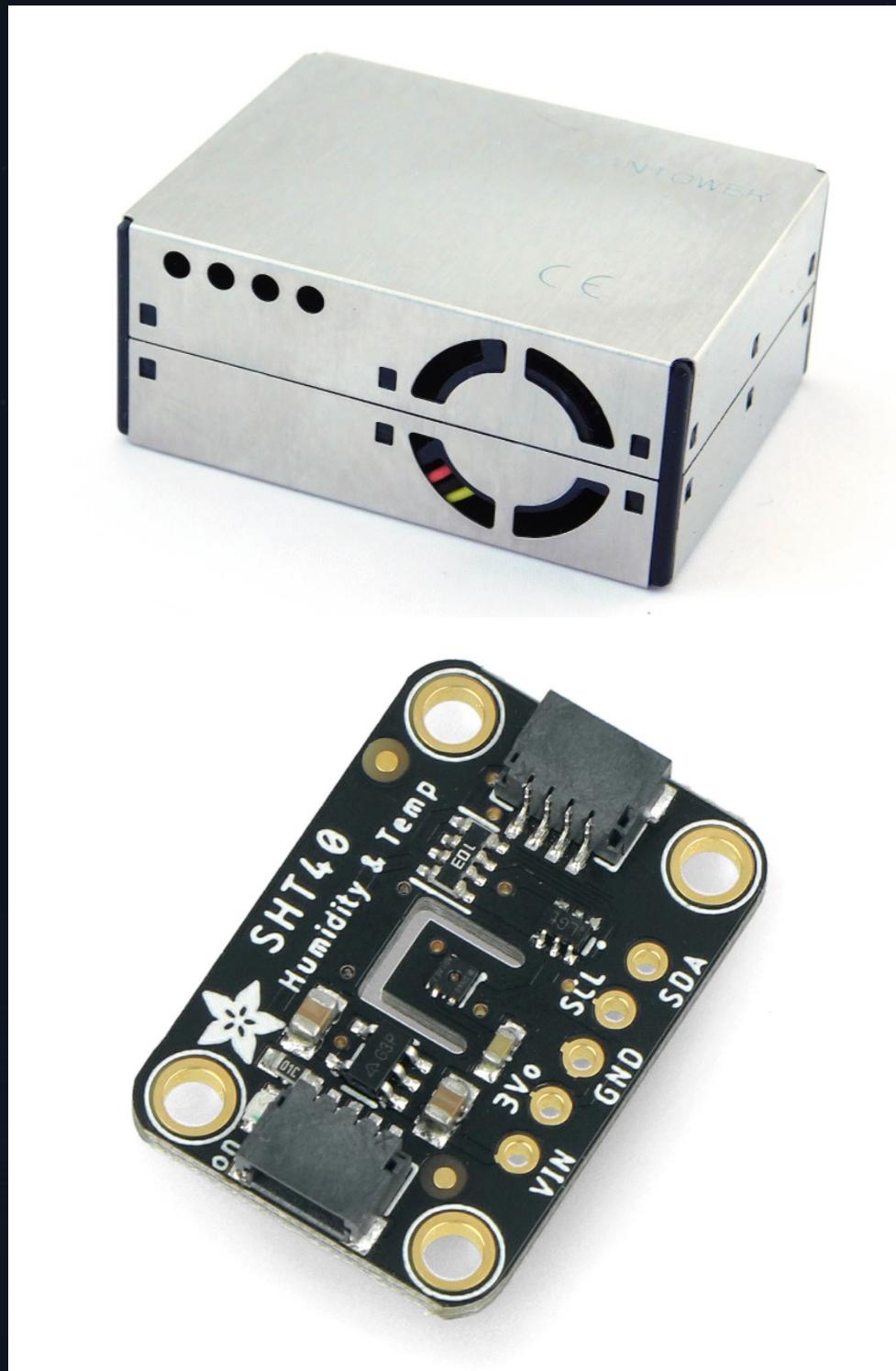
- Laserowy pomiar CO2.
- Technologia fotoakustyczna.
- Wysoka precyza, odporny na dryft.

## Czujnik BME688

- Inteligentny czujnik gazów (VOC).
- Pomiar ciśnienia, temperatury, wilgotności.

Wykorzystanie tych zaawansowanych sensorów gwarantuje rzetelne dane niezbędne do oceny jakości powietrza.





# Komponenty Hardware: Pyły i Termodynamika

Kluczowe czujniki do monitorowania cząstek stałych i mikroklimatu.

## Czujnik PMS5003 (Pyły PM)

- Laserowy licznik cząstek.
- Mierzy stężenie PM1.0, PM2.5, PM10.
- Niezbędny do wykrywania smogu i kurzu.

## Czujnik SHT40 (Temperatura, Wilgotność)

- Czwarta generacja od Sensirion.
- Niski pobór prądu.
- Wysoka odporność na kondensację.

Te komponenty zapewniają kompleksową ocenę środowiska, od zanieczyszczeń po komfort termiczny.

# Kolektor Danych ([collector.py](#))



Sercem systemu jest skrypt Pythonowy, efektywnie zarządzający odczytami z sensorów.

- **Wielowątkowość:** Moduł **threading** pozwala na asynchroniczne odczyty. Jeden wątek stale monitoruje PMS5003 (UART), podczas gdy główna pętla co 10 sekund odpytuje czujniki I2C.
- **Implementacja bloków try-except** zapobiega awariom całego systemu w przypadku błędów odczytu z pojedynczego sensora.
- **Optymalizacja:** Odczyty nie blokują magistrali I2C, co jest kluczowe dla ciągłości działania.

## Raspberry Pi 3B

Główna jednostka obliczeniowa z systemem Linux, odpowiedzialna za koordynację wszystkich operacji.

Zapewnia stabilne środowisko dla skryptów kolektora danych.

# Struktura Bazy Danych i Backend

Efektywne przechowywanie i udostępnianie danych dzięki bazie danych i API Flask.

## Flask API (`app.py`)

- Udostępnia zgromadzone dane w standardowym formacie JSON.
- Zapewnia szybką i łatwą komunikację z panelem front-endowym.

## Baza Danych: SQLite (`sensors.db`)

- Wybrana ze względu na lekkość i niskie wymagania zasobowe.
- Idealna dla platformy Raspberry Pi, eliminująca potrzebę zewnętrznego serwera SQL.

## Tabela `readings`

- Przechowuje kluczowe parametry: Timestamp, temperatura, wilgotność, CO2, pyły (PM1.0, PM2.5, PM10), voc i iaq.

	timestamp	temp	hum	co2	pm10	pm25	pm100	voc	iaq
1	2026-01-29 14:1...	22.51	58.65	1916	1	4	5	NULL	NULL
2	2026-01-29 14:1...	22.53	58.65	1926	1	4	4	NULL	NULL
3	2026-01-29 14:1...	22.52	58.66	1926	1	4	4	NULL	NULL
4	2026-01-29 14:1...	22.52	58.62	1932	1	4	4	NULL	NULL
5	2026-01-29 14:1...	22.52	58.63	1937	1	4	4	NULL	NULL
6	2026-01-29 14:1...	22.51	58.63	1943	1	4	4	NULL	NULL
7	2026-01-29 14:1...	22.54	59.13	1946	1	4	4	NULL	NULL
8	2026-01-29 14:1...	22.53	58.77	1959	1	4	4	NULL	NULL
9	2026-01-29 14:1...	22.53	59.03	1971	1	4	4	NULL	NULL
10	2026-01-29 14:1...	22.53	59.24	1974	1	4	4	NULL	NULL
11	2026-01-29 14:1...	22.54	58.84	2009	2	4	4	NULL	NULL
12	2026-01-29 14:1...	22.55	58.81	2056	2	4	4	NULL	NULL
13	2026-01-29 14:1...	22.56	59.98	2061	2	4	4	NULL	NULL
14	2026-01-29 14:1...	22.55	59.18	2065	2	4	4	NULL	NULL
15	2026-01-29 14:1...	22.54	59.1	2070	2	4	4	NULL	NULL

# Panel Użytkownika (Frontend)

Interaktywna wizualizacja danych w czasie rzeczywistym, dostępna poprzez intuicyjny dashboard.

## Technologie Webowe

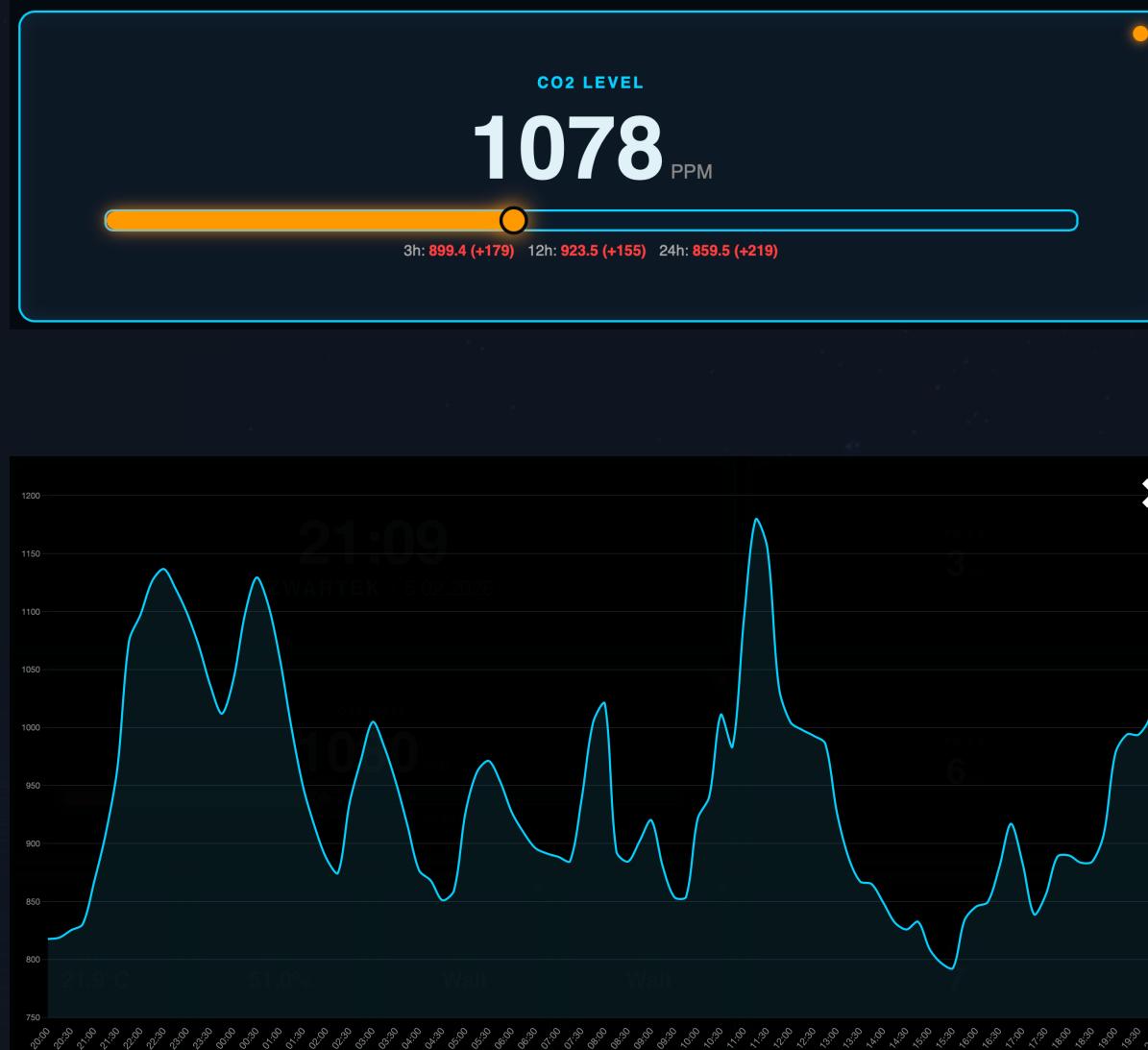
- HTML i CSS z wykorzystaniem Grid Layout dla responsywnego i estetycznego interfejsu.
- JavaScript do dynamicznego zarządzania treścią i interakcjami.

## Wizualizacja Danych

- Chart.js: Potężna biblioteka do renderowania wykresów liniowych, zapewniająca czytelną prezentację trendów.
- Dynamika Danych: Automatyczne odświeżanie danych co 5 sekund za pomocą AJAX/Fetch API, bez konieczności przeładowywania całej strony.



# Analiza Danych i Funkcje Specjalne



## Analiza Trendów (Funkcja `get_val_ago`)

Możliwość porównywania bieżących pomiarów z wartościami z przeszłości (np. 3 godziny temu), co pozwala na szybką ocenę dynamiki zmian jakości powietrza.

## Wygładzanie Danych

Implementacja średniej kroczącej w kodzie `app.py` wygładza wykresy, eliminując chwilowe fluktuacje i ułatwiając interpretację trendów długoterminowych.

# Napotkane Wyzwania i Rozwiązańia

- 1
- 2

## Konflikt Czasowy Odczytów

**Problem:** Odczyt wielu czujników jednocześnie, zwłaszcza z różnych magistrali, mógł prowadzić do kolizji i błędów danych.

**Rozwiązanie:** Implementacja `data_lock` w Pythonie. Mechanizm ten zapobiega jednoczesnemu zapisowi i odczytowi ze współdzielonej pamięci, zapewniając integralność danych.

## Kalibracja Czujników CO2

**Problem:** Czujniki CO2 wymagają okresowej kalibracji dla zachowania precyzji, co stanowiło wyzwanie w autonomicznym systemie.

**Rozwiązanie:** Wykorzystanie specjalnych komend inicjalizacyjnych SCD41 (np. `Stop/Start continuous measurement`), co pozwala na programowe zarządzanie kalibracją i stabilizację pomiarów.