

System monitoringu parametrów powietrza dla pomieszczeń mieszkalnych

Autor: Bartosz Lenartowicz

Streszczenie

Poniższy dokument zawiera opis wraz z dokumentacją projektu pozwalającego monitorować stan powietrza w pomieszczeniach. Projekt ten składa się z trzech części szczegółowo opisanych poniżej. W skład jego wchodzi aplikacja monitorująca parametry powietrza, API z możliwością zapisu parametrów do bazy danych i strona internetowa wyświetlająca zebrane wartości.

Spis treści

Streszczenie	2
Spis treści	2
Założenia	3
Schemat	3
Opis składowych	4
Urządzenia fizyczne	4
Oprogramowanie	4
System operacyjny	4
Odczyt wartości	4
API	5
GUI	6
Instalacja	7
Instalacja programu do odczytywania czujników	7
Instalacja aplikacji z API	7
Instalacja GUI	8
Podsumowanie i wnioski	9

Założenia

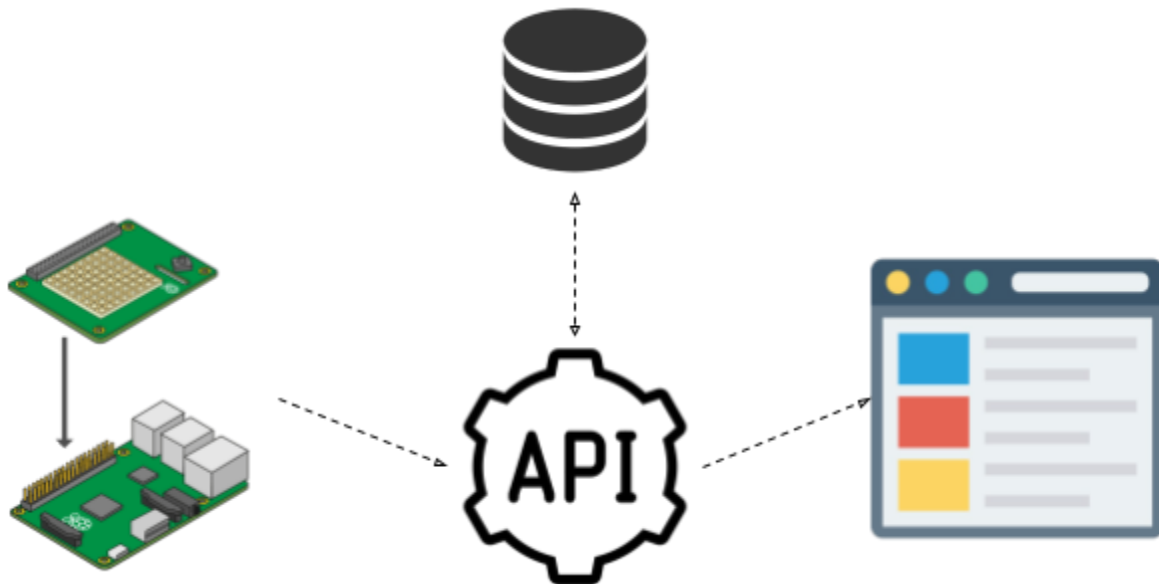
Podstawowym założeniem dla projektu było dostarczenie informacji o bieżącym stanie powietrza w pomieszczeniach mieszkalnych. Informacja powinna zawierać minimum trzy parametry:

- Temperature
- Wilgotność
- Ciśnienie

Wartości powinny być kolekcjonowane i dostarczane użytkownikowi w czasie rzeczywistym. Użytkownik powinien mieć możliwość ich łatwego podglądu. Dostarczone powinno być również API, z którego dane mogłyby być odczytywane w formacie json.

Schemat

Poniżej zaprezentowano schemat przepływu danych w projekcie. Linia ciągła oznacza komunikację przez interfejs fizyczny, a przerywana przez sieciowy. Po lewej stronie znajduje się urządzenie raspberry wraz z rozszerzeniem na którym znajdują się czujniki pomiaru temperatury. Następnie dane wysyłane są na serwer z zainstalowanym API. API zapisuje i pobiera dane do bazy. GUI aplikacji korzystając API wyświetla dane użytkownikowi.



Opis składowych

By zrealizować projekt należało dokonać szeregu wyborów urządzeń i technologii. Za główne kryterium do wyboru każdego elementu była prostota i szybkość w jego implementacji.

Urządzenia fizyczne

Projekt w swoich założeniach miał kolekcjonować rzeczywiste wartości dlatego należało wybrać urządzenia fizyczne zdolne do ich odczytu. Do realizacji tego projektu zdecydowano się wybrać urządzenie jakim jest RaspberryPi 3 b posiadające rozszerzenie SENSE HAT. Zestaw ten posiada ogromną zaletę jakim jest ułatwiona implementacja algorytmów dzięki dostarczanym bibliotekom. Na urządzeniu można zainstalować system operacyjny oraz serwer aplikacji do znacząco pozwala skonfigurować cały projekt jako jeden fizyczny element. Rozszerzenie SENSE HAT posiada zestaw czujników takie jak: termometr, higrometr, barometr wykorzystywane w projekcie lecz również żyroskop, akcelerometr i magnetometr oraz joystick i matrycę z LED RGB. W projekcie wykorzystywano tylko trzy pierwsze czujniki.

Oprogramowanie

Największa część pracy jaka została wykonana w projekcie była poświęcona zagadnieniom związanym z oprogramowaniem. Duża część oprogramowania została napisana w celu wykonania zadań związanych z projektem. Natomiast warto również przyjrzeć się tym elementom, które zostały wykorzystane jako oprogramowanie zewnętrzne. Opis oprogramowania będzie przedstawiany w taki sposób analogiczny do przesyłu wartości pomiarów od czujników do wyświetlania użytkownikowi. Szczegóły instalacji każdego programu podane zostaną w rozdziale poświęconym instalacji.

System operacyjny

RaspberryPi 3 b jest urządzeniem z posiadającym wydajny 64-bitowy procesor ARM-8 Cortex-A53 Broadcom BCM2837 o taktowaniu 1,2 GHz. Pozwala to na zainstalowanie systemu operacyjnego który znacznie ułatwia pracę z urządzeniem. Spośród wielu systemów jakie można zainstalować na został wybrany system Raspbian. Producent urządzenia zaleca wykorzystywanie tego systemu w przypadku pracy z układami wejścia-wyjścia, ponieważ umożliwia on korzystanie z wielu bibliotek do ich obsługi. Biblioteki te napisane są w języku python.

Odczyt wartości

Do zbadania wartości pomiarów wykorzystano rozszerzenie Sense-Hat dla RaspberryPi posiadające wbudowany barometr, higrometr i termometr. Twórcy tego urządzenia dostarczają również bibliotekę w języku python o takiej samej nazwie "sense-hat". Instalacja dodatku od użytkownika wymaga jedynie połączenia z Raspberry, instalacji biblioteki i restartu zestawu.

Pierwszy z programów napisanych w celu tego projektu miał na celu odczyt oraz przesłanie danych z czujników. Domyślnie odczyt następuje w nieskończonej pętli co 30 sekund. Po odczytu danych dane wysyłane są za pomocą komendy POST protokołu http w formacie JSON. Warto dodać, że czujniki ciśnienia i wilgotności również posiadają możliwość odczytu wartości temperatury dlatego zapisywany pomiar temperatury jest średnią wartością pomiarów z tych trzech czujników. Program również został napisany w taki sposób by zawsze zbierać wartości o pełnej lub idealnej połowy minuty

API

Drugim z programów stworzonym dla realizacji projektu był program obsługujący API aplikacji. Program został napisany w języku javascript z wykorzystaniem technologii nodejs. Dodatkowo użyto biblioteki express by ułatwić pracę z zapytaniami http. Program posiadał 4 adresy url. Adres źródłowy aplikacji pozwala obsługiwać zapytania typu POST w których przesyłane są dane z poprzedniego programu odczytującego wartości z czujników. Wykonanie zapytania GET pod tym adresem udostępnia 20 najnowszych wartości powietrza wraz z datą i czasem w formacie JSON. Dodatkowo udostępnione zostały 3 oddzielne ścieżki pod adresami:

- <<rootURL>>/temperature
- <<rootURL>>/humidity
- <<rootURL>>/pressure

skąd można odczytać ostatnie 20 wartości z jednego konkretnego czujnika wraz datą i czasem pomiaru. Wyświetlanie przykładowej wartości dostępnej pod adresem pokazano poniżej

```
{
  {
    "date": "13/01/2021",
    "time": "22:30:30",
    "temperature": 23.5,
    "humidity": 51.3,
    "pressure": 1124.1
  },{
    "date": "13/01/2021",
    "time": "22:30:30",
    "temperature": 23.5,
    "humidity": 51.3,
    "pressure": 1124.1
  }, (...)
}
```

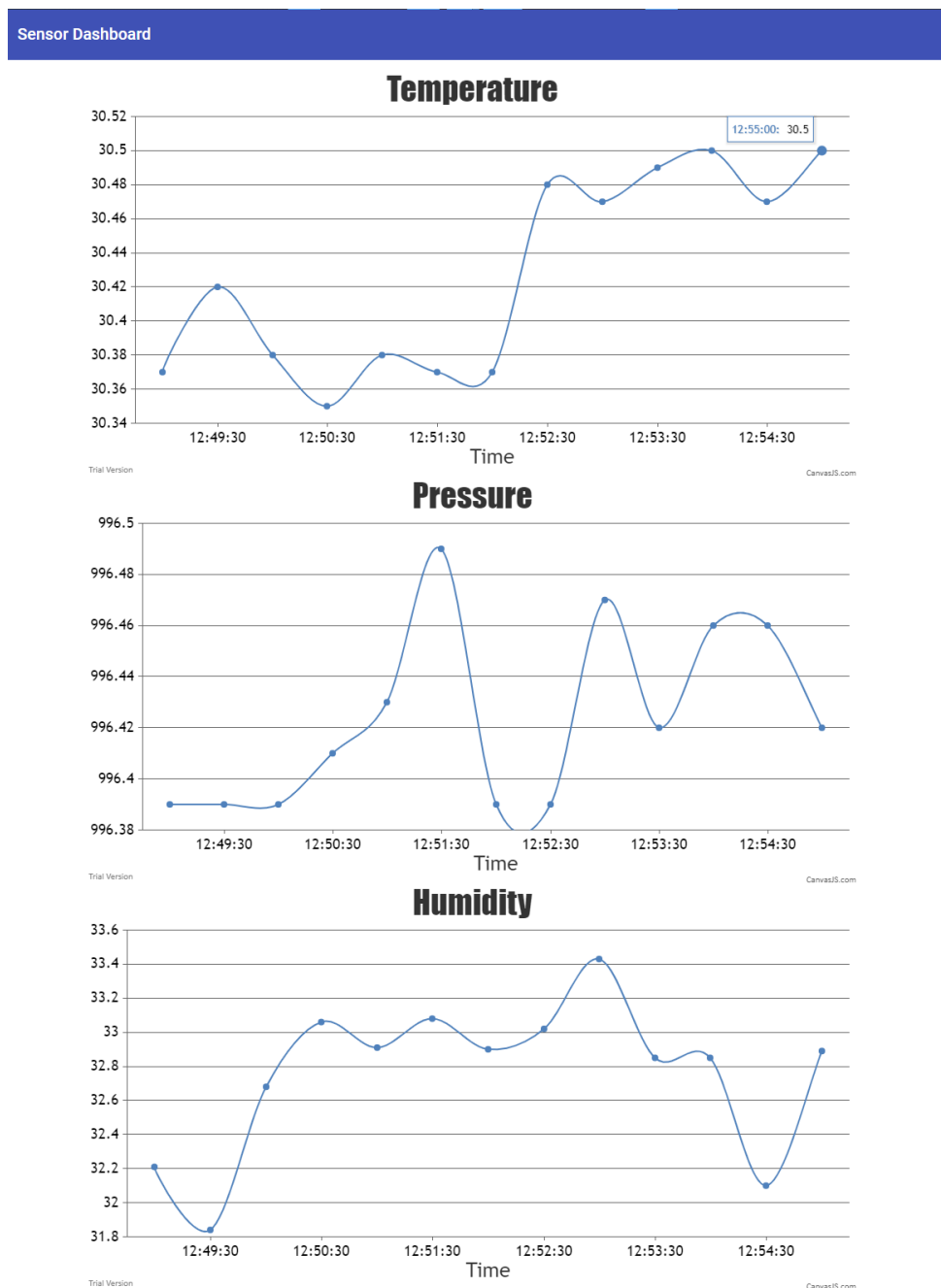
Do przechowania wartości wykorzystana została baza danych noSQL mongodb pozwalająca przechowywać obiekty w postaci pliku JSON.

Stworzono również możliwość wygodnej instalacji programu wraz z bazą danych jako kontenerów Docker'owych.

GUI

Miejszem z którego użytkownik może odczytać aktualne wartości jest interfejs użytkownika dostępny w przeglądarce internetowej. Interface został napisany w języku typescript oraz z wykorzystaniem frameworka Angular. Podobnie jak API dla wygody administratora aplikację można zainstalować jako kontener. Przy uruchamianiu kontenera należy podać adres URL poprzedniej aplikacji z API. Po otwarciu strony i wpisaniu adresu IP kontenera oczom ukazuje się zaprezentowana poniżej strona z 3 wykresami.

Interface użytkownika



Każdy wykres pokazuje 20 ostatnich pomiarów. Pomiary odświeżane są z tą samą częstotliwością, z którą dane uaktualniane są przez aplikację odczytującą pomiary z czujników. Po najechaniu myszą na konkretny punkt pokazany na wykresie ukazuje nam się dokładny wynik pomiaru.

Instalacja

Poniżej zaprezentowano instrukcję jaką należy wykonać by zainstalować programy. Do instalacji programów niezbędny jest dostęp do sieci internetowej z urządzeń na których będzie przebiegać instalacja oraz program git.

Instalacja programu do odczytywania czujników

By zainstalować program należy posiadać urządzenie RaspberryPi 3b z zainstalowanym systemem Raspbian oraz podłączonym rozszerzeniem Sense-Hat. Warto wspomnieć, że konfiguracja i działanie wymagają podłączenia do sieci internetowej. Wpierw należy pobrać niezbędne biblioteki języka python poleceniem wykonanym w terminalu:

→ `pip install sense-hat request`

Następnie należy pobrać program `sensors_values_reader` z repozytorium i go zainstalować za pomocą poniższych komend. Wartość `<<API_URL>>` to adres url serwera na którym będzie zainstalowana aplikacja z API.

→ `git fetch https://github.com/Bartosz95/sensors_values_reader.git`

→ `sudo chmod +x installation.sh`

→ `sudo ./installation.sh <<API_URL>>`

I uruchomić urządzenie ponownie co kończy proces instalacji pierwszej aplikacji. Od tego momentu automatycznie przy starcie systemu uruchamiany będzie również program wysyłający dane na podany server. W celu odinstalowywanie programu należy uruchomić skrypt `uninstall.sh` poźrównież z prawami administratora.

Instalacja aplikacji z API

Do zainstalowania drugiej w kolejności aplikacji potrzebne jest również urządzenie z dostępem do internetu. Można wykorzystać RaspberryPi użyte do odczytu wartości z czujników lub inny serwer aplikacji. Server ten powinien mieć możliwość uruchomienia aplikacji w dockerze oraz program `docker-compose`. W celu instalacji należy pobrać repozytorium git i uruchomić aplikację

→ `git fetch https://github.com/Bartosz95/sensors_api.git`

→ `docker-compose up`

Od teraz jest możliwość wykonania zapytania POST na adres serwera co pozwala skolekcjonować dane. Przykładowe zapytanie pokazane jest poniżej.

→ `curl -XPOST -H "Content-Type: application/json" {"date": "13/01/2021", "time": "22:30:30", "temperature": 23.5, "humidity": 51.3, "pressure": 1124.1} http://localhost:3000/api/v1`

Zebrane dane można zobaczyć zebrane w przeglądarce pod adresem aplikacji według poniższego schematu:

- `http://localhost:3000/api/v1/`
- `http://localhost:3000/api/v1/temperature`
- `http://localhost:3000/api/v1/humidity`
- `http://localhost:3000/api/v1/pressure`

Lub za pomocą metody GET wykonanej w inny sposób np za pomocą programu curl. Przykład poniżej

→ `curl http://localhost:3000/api/v1/temperature`

Instalacja GUI

Analogicznie jak instalacja API aplikacja gui wymaga programu docker oraz docker-compose. Ponadto wymagany jest adres API. Wpierw należy pobrać repozytorium

→ `git fetch https://github.com/Bartosz95/sensors_gui.git`

Następnie w pliku `nginx.conf` należy uzupełnić pole `<<API_URL>>` adresem pod którym dostępne są dane z aplikacji API. Przykładowy wygląd pliku pokazany jest poniżej.

```
server {  
    listen 80;  
    root /usr/share/nginx/html;  
    index index.html;  
  
    location /api/v1/ {  
        proxy_pass http://localhost:3000/api/v1/ ;  
    }  
}
```

Konfigurację kończy uruchomienie aplikacji za pomocą polecenia:

→ docker-compose up

W tym momencie pod adresem URL serwera na którym zainstalowane jest GUI powinna pokazać się strona z wykresami temperatur. Domyślnie co 30 sekund wykresy będą się odświeżać i pokazywać aktualne wartości.

Podsumowanie i wnioski

Aplikacja stworzona w celu wykonania projektu spełnia swoje podstawowe zadanie - wyświetla informację o bieżącym stanie temperatury. Programy wchodzące w skład projektu zostały napisane z użyciem najnowszych dostępnych darmowych technologii co znacznie przyspieszyło i poprawiło komfort pracy przy projekcie. Każdy z komponentów jest napisany w taki sposób by umożliwić łatwe jego skalowanie w przyszłości. Proces instalacyjny aplikacji również został bardzo uproszczony.

Niestety nie wszystkie elementy udało się zrealizować według pierwotnego zamysłu. Pochylić się należy nad brakiem zabezpieczeń w aplikacji. Obecnie każdy może wykonywać zapytania na serwer które tworzą użytkownik może nie tylko bez problemu podejrzeć dane lecz również dodać nowe. Zapewnienie bezpieczeństwa aplikacji obowiązkowo powinno być wprowadzone jak najszybciej, a już na pewno przed wcześniejszym zainstalowaniem aplikacji na serwerach publicznych. Drugą rzeczą nad którą można się pochylić to wprowadzenie bardziej rozbudowanego interface'u użytkownika z możliwością regulacji parametrów. W planach istnieje również przystosowanie aplikacji do jej instalacji za pomocą technologii Kubernetes. W tym celu projekt został już umieszczony jako zestaw kontenerów w platformie Docker Hub. Ostatnią rzeczą, którą można poruszyć jest przyszła rozbudowa projektu o kolejne urządzenie pozwalające ingerować w temperaturę pomieszczeń. Grzałka z możliwością zewnętrznej regulacji pozwoliłaby stworzyć układ, do regulacji temperatury który w rzeczywisty sposób mógłby przynieść korzyść użytkownikowi.

Podsumowując należy stwierdzić, że projekt został wykonany w poprawny sposób, a co ważniejsze ma duże możliwości dalszej jego rozbudowy.