

Wyższa Szkoła Bankowa w Toruniu
Wydział Finansów i Zarządzania w Bydgoszczy
Studia podyplomowe
Kierunek: Programista Python Developer

Patryk Łuczak

21585

**Aplikacja webowa połączona ze stacją pogodową
opartą o mikrokontroler ESP32**

Praca dyplomowa

Bydgoszcz, 2023 r.

Praca napisana pod kierunkiem

dr inż. Mariusz Mol

Spis treści

1. Cel projektu.....	2
2. Warstwa sprzętowa	3
3. Aplikacja webowa.....	5
4. Tendencje rozwojowe	6
5. Podsumowanie	7

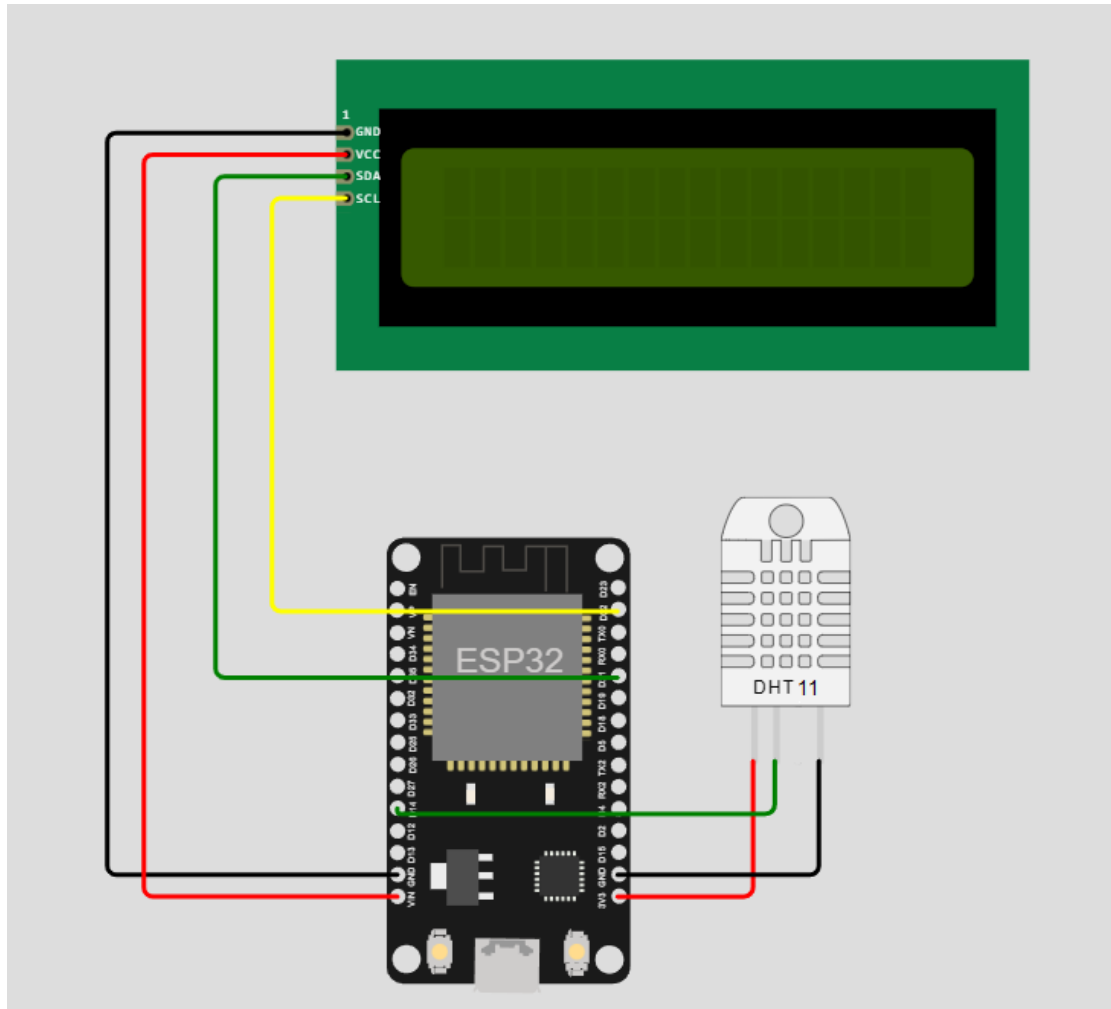
1. Cel projektu

Celem projektu było zbudowanie aplikacji webowej opartej o framework *Django* połączonej ze stacją pogodową opartą o mikrokontroler ESP32. Załoženiami dla stacji pogodowej było: cykliczny pomiar temperatury oraz wilgotności w pomieszczeniu przy użyciu czujnika DHT-11, wyświetlenie wyników pomiarów na wyświetlaczu LCD oraz przesłanie wyników na serwer przy użyciu REST API. W przypadku aplikacji webowej założeniem było stworzenie interfejsu umożliwiającego otrzymanie danych ze stacji pogodowej, możliwość zarządzania tymi danymi oraz wyświetlanie aktualnego odczytu. Dodatkowym zadaniem aplikacji miało być wyświetlanie aktualnej pogody w miejscowości wpisanej w odpowiednim polu tekstowym oraz możliwość tworzenia nowych użytkowników, logowania się i wylogowania. Z poziomu panelu administracyjnego zakładano możliwość zarządzania użytkownikami.

Wszystkie te założenia zostały spełnione, jednak podczas tworzenia projektu powstało parę ciekawych tendencji rozwojowych, które zostały opisane w dalszej części tego opisu. Aplikacja internetowa została napisana w środowisku *Visual Studio Code*, natomiast program dla ESP32 został napisany przy użyciu *Thonny IDE*, który oferował wsparcie dla niektórych mikrokontrolerów jak np. *Raspberry Pi Pico* czy właśnie *ESP32*. Kod dla platformy sprzętowej został napisany w języku *MicroPython*, natomiast część webowa w języku *Python* przy wykorzystaniu framework'a *Django* i *Django REST framework*. Baza danych użyta w projekcie to *SQLite*.

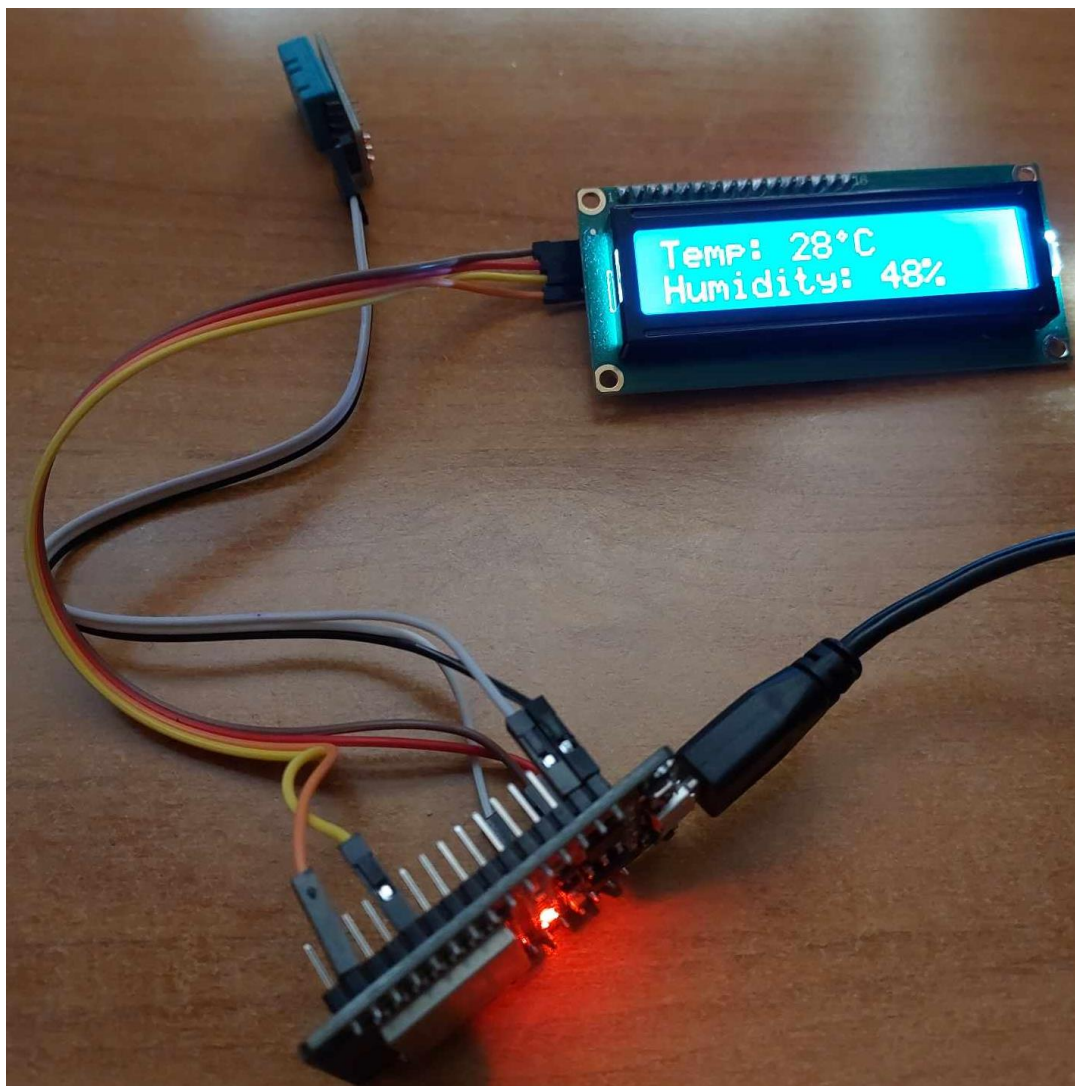
2. Warstwa sprzętowa

Obwód elektryczny został oparty o mikrokontroler *ESP-WROOM-32*, czujnik temperatury i wilgotności *DHT11* oraz wyświetlacz LCD z konwerterem magistrali I2C.



Rys. 2.1. Schemat elektryczny stacji pogodowej

W celu poprawnego działania układu obwód należy połączyć zgodnie ze schematem na rysunku 2.1. Piny *SDA* oraz *SCL* wyświetlacza LCD należy połączyć kolejno z pinami 21 oraz 22 mikrokontrolera, natomiast środkową nóżkę czujnika *DHT11* z pinem nr 14. Pozostałe wyprowadzenia to zasilanie peryferiów, przy czym należy pamiętać, że czujnik zasilany jest z napięcia równego 3,3 V.

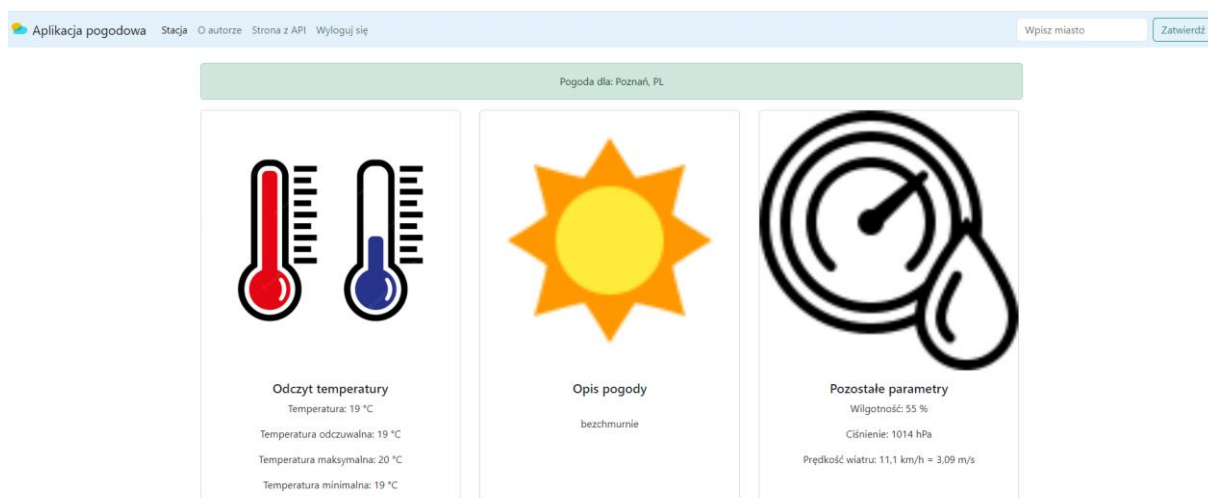


Rys. 2.2. Fizyczny układ w trakcie działania

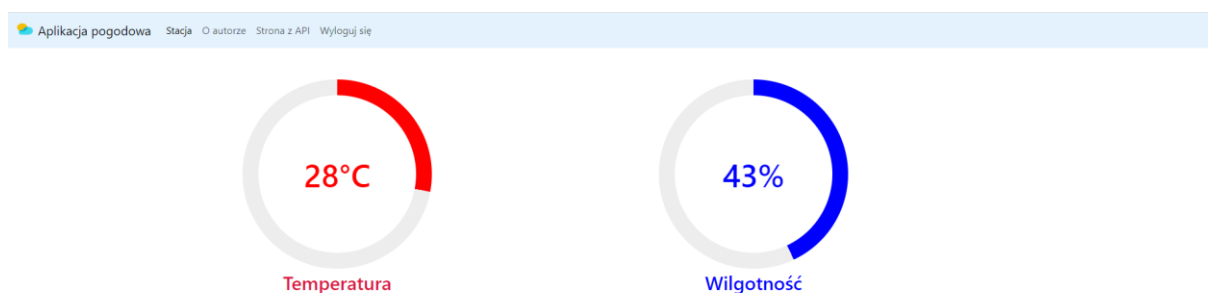
Na zdjęciu (rys.2.2) został przedstawiony fizyczny układ po wykonaniu pomiaru. Firmware działający w układzie został podzielony na funkcje. Przed rozpoczęciem pętli głównej zostaje wywołana funkcja odpowiedzialna za połączenie z domową siecią Wi – Fi, a w pętli głównej zostaje wykonany pomiar, następuje wyświetlenie danych na wyświetlaczu i wysłanie ich na serwer. Czynności te są wykonywane cyklicznie co około minutę. Przy wyświetlaniu danych na wyświetlaczu wykorzystano gotowe biblioteki *i2c_lcd.py* oraz *lcd_api.py*.

3. Aplikacja webowa

Zaprojektowana strona została stworzona we framework'u *Django* przy pomocy technologii webowych takich jak: *HTML*, *CSS* i *Javascript*. Umożliwia ona tworzenie kont użytkowników, logowanie i wylogowanie się. Składa się z dwóch głównych funkcjonalności, pierwsza to wyświetlenie danych pogodowych z miasta wpisanego w polu tekstowym przy użyciu API ze strony <https://openweathermap.org/> (rys. 3.1). Druga aplikacja odpowiada za wyświetlenie danych pomiarowych z *ESP32* (rys. 3.2), jednak aby wejść w tę zakładkę należy zalogować się.



Rys. 3.1. Aplikacja z danymi pobranymi przez API



Rys. 3.2. Aplikacja z danymi pomiarowymi

Dane pomiarowe są odbierane przez serwer dzięki wykorzystaniu *Django Rest Framework*.

4. Tendencje rozwojowe

W czasie tworzenia programów pojawiły się dodatkowe pomysły na rozwinięcie projektu. W przypadku warstwy sprzętowej można dołożyć czujnik *BMP280*, który dodatkowo umożliwia pomiar ciśnienia. Można również zastosować serwomechanizmy, które sterowałyby pewnym elementem wykonawczym. W tym przypadku to układ elektroniczny odbierałby dane z serwera i na tej podstawie sterował serwomechanizmami. Elementem wykonawczym mogłaby być kamera wraz z *ESP32 – Cam*, która przesyłałaby obraz na serwer. Można tutaj pokusić się o zbudowanie modelu uczenia maszynowego, którego zadaniem byłoby rozpoznawanie określonych przedmiotów lub twarzy. Układ można rozszerzyć o przyciski i stworzyć pewnego rodzaju menu na wyświetlaczu LCD, umożliwiałoby to przesuwanie wyświetlanych danych w momencie, gdyby było ich więcej, a nawet sterowanie elementami wykonawczymi również z poziomu przycisków.

Część webowa może zostać rozwinięta o dodanie mapy, która wyświetlałaby pinezkę z miastem wpisanym w polu tekstowym. Tutaj też można skorzystać z dostępnych API. Aplikacja pomiarowa powinna umożliwiać wyświetlanie kilku lub kilkunastu poprzednich pomiarów z bazy danych w formie tabelarycznej lub na wykresie. Spowodowałoby to lepszą analizę danych pomiarowych. Można również rozszerzyć sieć czujników o więcej pomieszczeń i wyświetlać te dane osobno dla każdego z nich. Dostosowując stronę do sprzętu, należałoby stworzyć interfejs graficzny dla sterowania serwomechanizmami i miejsce na widok z kamery. Aplikacja mogłaby również generować raporty dzienne do plików *xls* z danymi pomiarowymi.

5. Podsumowanie

Wszystkie założenia zostały spełnione, jednak ze względu na fakt, że projekt ma potencjał do rozwoju prawdopodobnie zostanie jeszcze ulepszony i rozwinięty. Projekt w pewnym stopniu pokazuje możliwości języków *Python* czy *MicroPython* oraz framework'a *Django*. Pokazuje też jak w stosunkowo prosty sposób zaprogramować komunikację przez sieć między dwoma urządzeniami, które wymieniają się danymi. Rodzi to pomysły na różne rozwiązania dla takiego układu jakim jest *ESP32*. Można zbudować system automatyki domowej wraz z systemami alarmowymi, dzięki którym użytkownik mógłby w sieci przeglądać stan włączników w mieszkaniu, odczyty z czujników i powiadomienia z alarmów oraz mógłby sterować tymi włącznikami czy innymi urządzeniami wykonawczymi. Innym pomysłem jest robot sterowany zdalnie przez sieć, który mógłby być mobilną stacją pogodową. Jak widać możliwości jest wiele, dlatego warto rozwijać swoje umiejętności i wiedzę w zakresie programowania w języku *Python*.