

OPERATING SYSTEMS FOR EMBEDDED SYSTEMS



AGH

AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA
W KRAKOWIE

Projekt Monitoring parametrów rozwoju roślin; czujniki, WiFi, RTOS

Autorzy: Jakub Mirola, Bartosz Wójcik

Spis treści

1. [Założenia projektowe](#)
2. [Struktura](#)
3. [Metodyki projektowania](#)
4. [Implementacja](#)
5. [Podsumowanie](#)
6. [Bibliografia](#)

1. Założenia projektowe

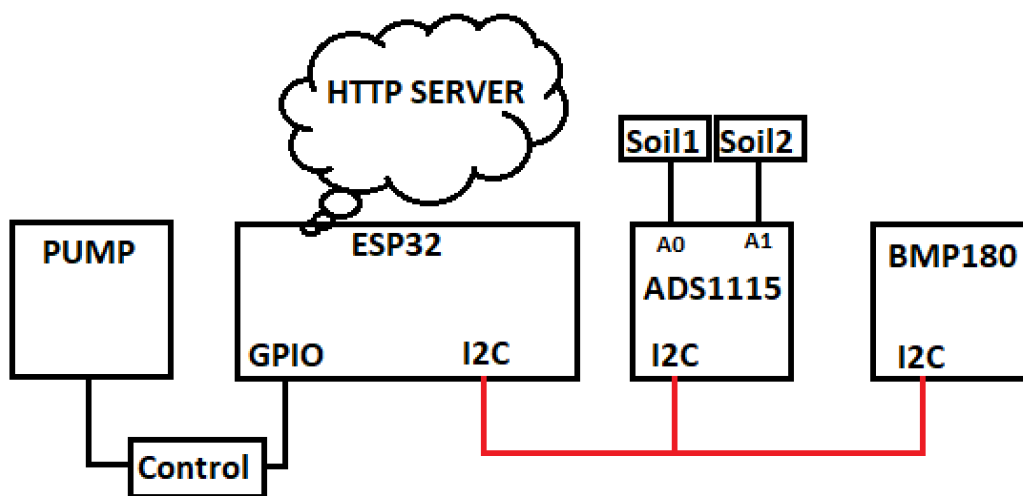
Celem projektu jest stworzenie systemu do monitorowania wilgotności gleby, w której posadzone są rośliny. Urządzenie ma na bieżąco monitorować pobrane dane z czujników i zapisywać je na serwerze HTTP. Użytkownik w przypadku gdy wilgotność osiągnie zbyt niski poziom będzie mógł uruchomić pompę wodną. Użycie w projekcie FreeRTOS umożliwi jednocześnie sterowanie pompą, uruchomienie serwera oraz pobieranie danych z czujników.

Założenia do projektu:

- Server HTTP na ESP32
- Statyczny adres IP
- Pomiar danych z czujników: wilgotność gleby, bmp180
- Sterowanie pompą wody
- Użycie FreeRTOS

2. Struktura

Na rysunku numer 1 został przedstawiony schemat ideowy projektu. Jako główny element zdecydowano o użyciu ESP32. Ten mikrokontroler to seria niedrogich, energooszczędnych układów z wbudowanym Wi-Fi i Bluetooth. Są wersje dwurdzeniowe oraz jednordzeniowe. W tym projekcie została użyta wersja dwurdzeniowa. Jedną z wielu zalet ESP32, oprócz wifi i Bluetooth jest duża liczba dostępnych innych peryferiów takich jak I2C, SPI, 12 bitowy ADC, 8 bitowy DAC, I2S, UART, PWM, kontroler kart SD. Ta gama dostępności pozwala na zastosowanie ESP32 w praktycznie każdym projekcie IoT.



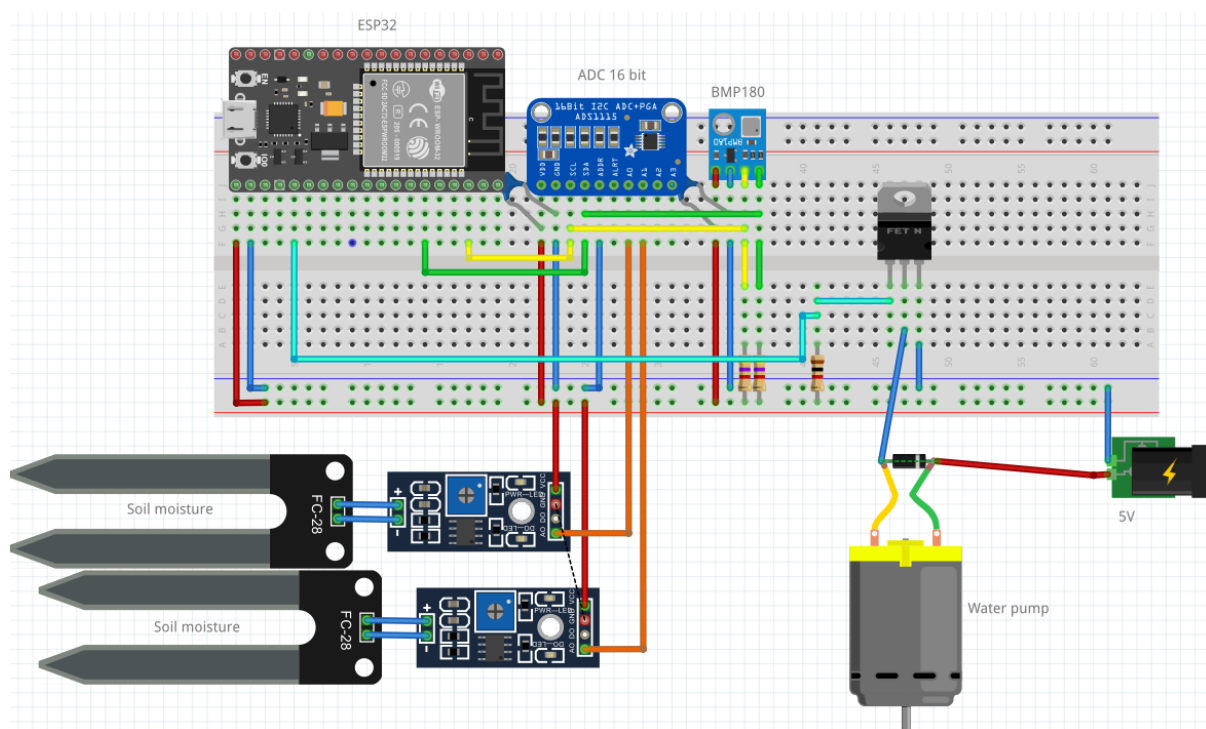
Rys. 1 Schemat ideowy

Celem projektu było monitorowanie stanu roślin. W tym celu zdecydowano się na użycie czujników wilgotności gleby jako że jest to podstawowy i ważny aspekt przy hodowaniu roślin. Najprostsze czujniki wilgotności są tanie i posiadają interfejs analogowy. Aby pomiary z czujników były dokładniejsze postanowiono skorzystać z zewnętrznego modułu przetwornika analogowo cyfrowego ADS1115. Jest to bardzo mały, energooszczędny, 16-bitowy precyzyjny przetwornik z wewnętrznym napięciem odniesienia. Stosowany głównie w precyzyjnym oprzyrządowaniu. Jego funkcją jest wzmacnianie, poprawianie dokładności i konwersja sygnału analogowego na cyfrowy w celu analizy danych.

Kolejnym czujnikiem, który został użyty jest BMP180. Czujniki z tej serii są przeznaczone do pomiaru ciśnienia barometrycznego lub ciśnienia atmosferycznego. BMP180 to precyzyjny czujnik przeznaczony do zastosowań konsumenckich. Dodatkowo czujnik ten mierzy temperaturę i wysokość nad poziomem morza.

Ostatnim ważnym elementem jest pompa i obwód sterowania do niej. Obwód sterowania do pompy można zobaczyć na rysunku numer 2. Jest to obwód z tranzystorem jako przełącznik. Dzięki temu można bez obaw włączyć lub

wyłączyć pompę wody. W początkowych założeniach nie było uwzględnionej pompy, natomiast wprowadziła dodatkową funkcjonalność do urządzenia. Dzięki temu oprócz monitoringu parametrów rośliny, można również zmieniać wilgotność gleby poprzez załączanie pompy. To pozwoli zachować roślinę w dobrej kondycji.

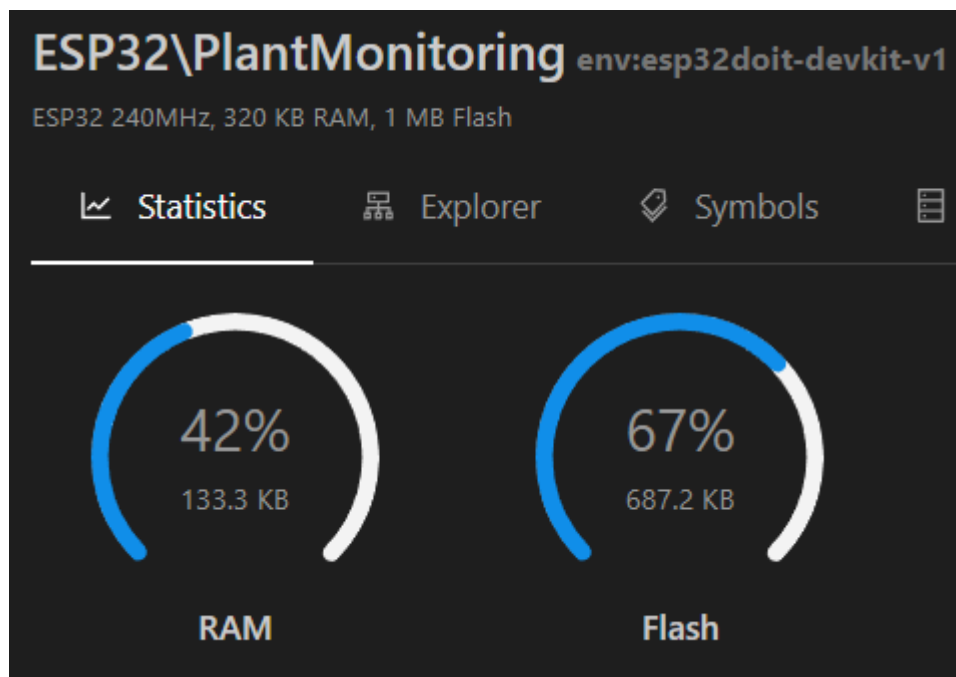


Rys. 2 Podłączenie elementów na płytce stykowej

Na rysunku numer 2 został umieszczony schemat podłączenia elementów na płytce stykowej. Dzięki użyciu I2C ilość przewodów jest ograniczona, ponieważ BMP180 oraz ADS1115 używają tego interfejsu do komunikacji. Należy jednak zapewnić aby adresy tych urządzeń nie były takie same. Na rysunku numer 2 widać dodatkowo, że do sterowania pompą jest potrzebne zewnętrzne zasilanie. Wynika to z faktu iż pompa potrzebuje co najmniej 5V zasilania a samo ESP32 potrzebuje do działania 3.3V.

Użyte narzędzia do stworzenia projektu:

- **PlatformIO** - to ekosystem typu open source do opracowywania projektów IoT (Internet of Things). Jest to wieloplatformowe narzędzie, które zapewnia ujednolicony system kompilacji dla różnych platform i frameworków, takich jak Arduino, Raspberry Pi, ESP32 i inne. PlatformIO integruje się z popularnymi zintegrowanymi środowiskami programistycznymi (IDE), takimi jak Visual Studio Code, Eclipse i inne, oraz zapewnia różnorodne funkcje do opracowywania i debugowania projektów IoT. Dodatkowo umożliwia inspekcje kodu oraz importowanie potrzebnych bibliotek.



Rys. 3 Funkcja inspekcji w PlatformIO

- **Trello** - opisane w rozdziale 3

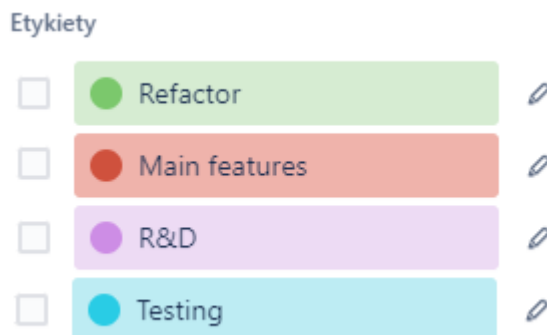
- **GitHub** - to internetowa platforma służąca do kontroli wersji i współpracy przy projektach oprogramowania. Opiera się na systemie kontroli wersji Git, który umożliwia programistom śledzenie zmian w ich kodzie i współpracę z innymi członkami zespołu. GitHub zapewnia oparte na chmurze repozytorium do przechowywania kodu i zarządzania nim, a także narzędzia do przeglądania i omawiania zmian w kodzie. Jest szeroko stosowany przez programistów i zespoły do hostowania i zarządzania projektami i jest często używany w projektach open source.

- **Visual Studio Code** - to darmowy edytor kodu typu open source opracowany przez firmę Microsoft dla systemów Windows, Linux i macOS. Jest to lekki, ale potężny edytor kodu, który jest powszechnie używany przez programistów do tworzenia aplikacji, stron internetowych oraz wielu różnych projektów.

3. Metodyki projektowania

Podczas tworzenia projektu wykorzystano narzędzie Trello do koordynowania pracy nad projektem. Trello to internetowe narzędzie do zarządzania projektami i współpracy, które umożliwia zespołom organizowanie i ustalanie priorytetów ich pracy. Opiera się na koncepcji „tablic”, które służą do przechowywania list „kart”. Każda karta reprezentuje zadanie lub część pracy i może być przenoszona między listami w miarę postępu pracy. Trello zapewnia wizualny interfejs do zarządzania projektami i może być używany do różnych celów, takich jak śledzenie postępów projektów tworzenia oprogramowania, zarządzanie kampaniami marketingowymi i organizowanie zadań osobistych.

Zdecydowano się na użycie Trello z kilku powodów, przede wszystkim zapewnia wizualny interfejs do organizowania i ustalania priorytetów pracy, co może być bardziej efektywne niż używanie tekstowej listy rzeczy do zrobienia. Możliwość przenoszenia kart między listami i szybkiego podglądu postępu prac może pomóc zespołom w utrzymaniu organizacji i podążaniu zgodnie z planem. Drugim przydatnym aspektem Trello jest możliwość zapraszania członków zespołu do swoich tablic i współpracę nad projektami w czasie rzeczywistym. Dodatkowo funkcja komentarzy i wzmianek ułatwiają komunikację i są na bieżąco z postępem prac. Kolejną przydatną funkcją Trello są różnorodne opcje dostosowywania, takie jak możliwość dodawania etykiet, naklejek i niestandardowych tła do tablic. Może to ułatwić organizowanie i ustalanie priorytetów pracy oraz personalizację korzystania z Trello. Ostatnią, ale równie ważną cechą Trello jest łatwość użytkowania. Ma prosty, intuicyjny interfejs, co sprawia, że jest to dobry wybór dla zespołów o różnej wielkości i poziomie doświadczenia.



Rys. 4 Nazwane oraz pokolorowane etykiety użyte w projekcie

Etykiety zobrazowane powyżej zostały podzielone na 4 typy.

- **Refactor** - Refaktoryzacja to proces modyfikowania i ulepszania projektu istniejącego kodu bez zmiany jego funkcjonalności. Jest to powszechna praktyka w tworzeniu oprogramowania i często ma na celu poprawę łatwości konserwacji, czytelności i wydajności kodu. Refaktoryzacja może obejmować reorganizację kodu, zmianę nazw zmiennych i funkcji oraz zmianę struktury

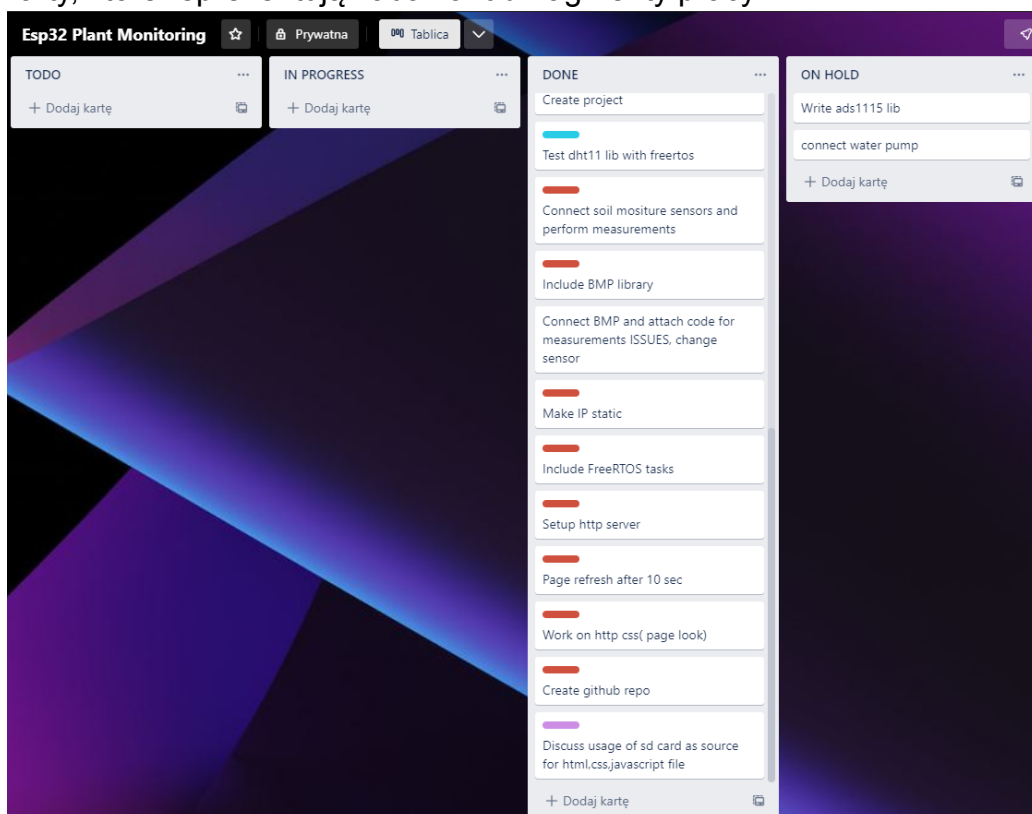
kodu. Jest to ważna część procesu tworzenia oprogramowania, ponieważ pomaga utrzymać czystość i łatwość zrozumienia bazy kodu, co może ułatwić utrzymanie i dodawanie nowych funkcji do bazy kodu w miarę upływu czasu.

- **Main features** - Główne cechy projektu, zadania o najwyższej wadze, które musiały być realizowane priorytetowo

- **R&D** - Badania i rozwój są ważną częścią procesu innowacji i często obejmują przeprowadzanie eksperymentów, testowanie prototypów i gromadzenie danych w celu tworzenia nowych pomysłów lub ulepszania istniejących

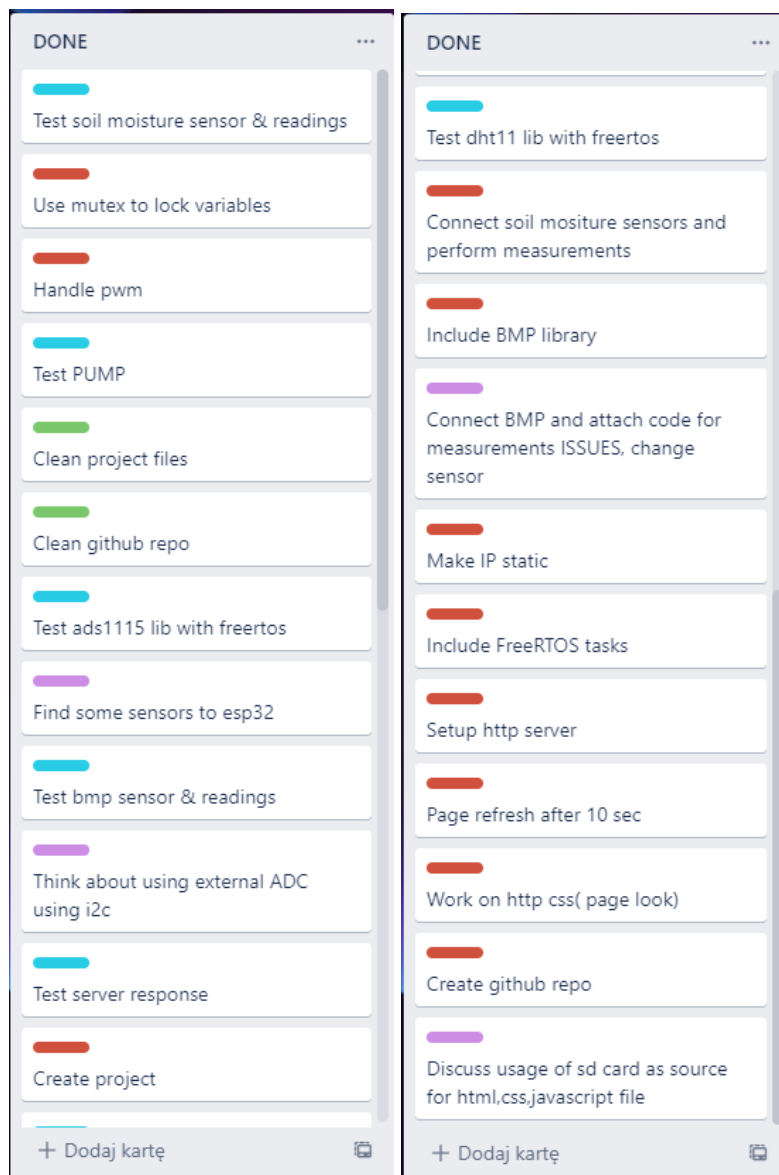
- **Testing** - Testowanie jest ważnym krokiem w procesie tworzenia oprogramowania. Jest to proces oceny systemu lub jego komponentów w celu ustalenia, czy spełnia on określone wymagania, czy też nie. Testowanie ma na celu upewnienie się, że tworzone oprogramowanie jest wysokiej jakości i spełnia potrzeby użytkownika.

W tym projekcie stworzono listy przedstawiające różne etapy pracy, takie jak „Do zrobienia”, „W toku”, „Gotowe” i „Wstrzymane”. Po utworzeniu list dodano karty, które reprezentują zadania lub fragmenty pracy.



Rys. 5 Wygląd tablicy w Trello

W czasie pracy nad projektem dodawano kolejne zadania. Członkowie zespołu dzielili się pracą. Pełną listę zrealizowanych zadań można obejrzeć na poniższych obrazkach.



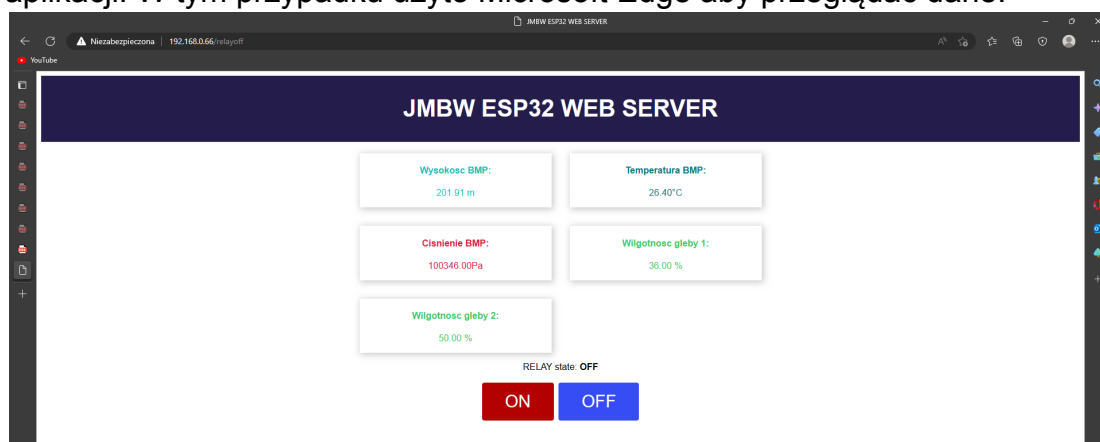
Rys. 6 Lista zrealizowanych zadań

Tworzone zadania w swoim założeniu miały być podzielone na mniejsze moduły składające się w całość. Podczas pracy nie mogło zabraknąć czasu na pisanie kodu, refaktoryzacje ale i oczywiście też na testy. Dzięki podziałowi pracy, wszystkie zadania mogły zostać zrealizowane. Ostatecznie Trello okazało się prostym, aczkolwiek skutecznym narzędziem do organizacji pracy. W przyszłości jeśli praca będzie prowadzona w małych zespołach oraz projekt nie będzie zbyt wielki, zostanie rozważone użycie tego narzędzia. W innych przypadkach natomiast raczej zostałyby użyte narzędzie Jira albo Azure Dev Ops.

4. Implementacja

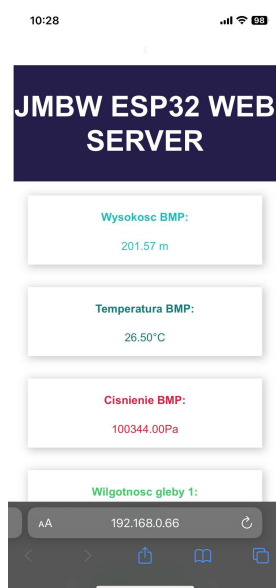
Tematem projektu była implementacja monitoringu parametrów rozwoju roślin, używając czujników. Dodatkowo w założeniach miał zostać użyty system operacyjny czasu rzeczywistego, oraz protokół WiFi.

Zdecydowano zatem aby zastosować server HTTP na ESP32. Dzięki temu jest możliwość monitorowania rośliny wewnątrz domowej sieci. Na poniższym obrazku umieszczono zdjęcie przedstawiające wygląd strony zbudowanej aplikacji. W tym przypadku użyto Microsoft Edge aby przeglądać dane.



Rys. 7 Wygląd strony internetowej do monitorowania parametrów rośliny używając komputera i Microsoft Edge

Dzięki zaimplementowaniu serwera HTTP możliwe jest przeglądanie danych oraz sterowanie pompą z dowolnego urządzenia posiadającego dostęp do WiFi oraz przeglądarkę internetową.

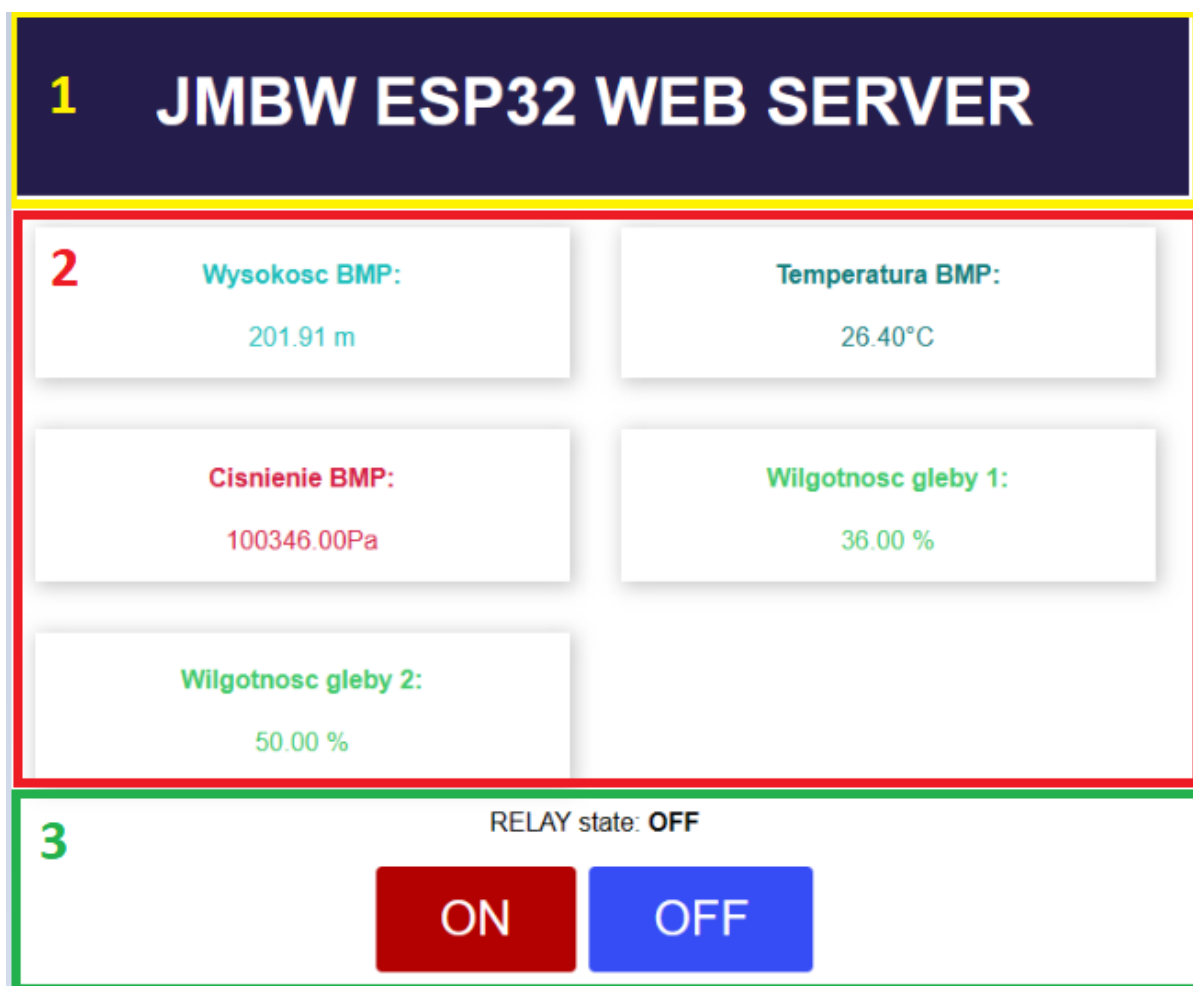


Rys. 8 Wygląd strony internetowej do monitorowania parametrów rośliny używając telefonu i przeglądarki Safari

Strona internetowa jest podzielona na 3 zasadnicze części. Pierwszą z nich oznaczoną na żółto jest tytuł strony.

Następna pokolorowana na czerwono jest strefa pomiarów. Przy pomocy CSS utworzono style aby wyniki pomiarów były przedstawione w przejrzystej i ładnej formie. Kaskadowe arkusze stylów (CSS) to język arkuszy stylów używany do opisywania wyglądu i formatowania dokumentu napisanego w HTML. Może być używany do kontrolowania koloru, czcionki, układu i innych stylów strony internetowej. W tej sekcji można zatem znaleźć dane z czujników BMP180 oraz czujników wilgotności gleby, które zostały odpowiednio podzielone i pokolorowane.

Trzecią i ostatnią częścią jest pokolorowana na zielono strefa sterowania pompą wody. Umieszczono tutaj obecny stan pompy(Może być on lub off), oraz 2 przyciski do sterowania nią.



Rys. 9 Podział strony na strefy

Dzięki stworzeniu paru stylów za pomocą CSS, jest możliwość skorzystania z nich w przyszłości aby rozbudować projekt o jeszcze większą funkcjonalność. Zastosowanie takich stylów pozwala na lepszą wizualizację danych przez co łatwiej i szybciej można przyswoić dane.

Podczas pracy nad projektem korzystano z dokumentacji ESP32 oraz wytycznych producenta co do programowania tego mikrokontrolera. Przydatne linki można znaleźć w rozdziale numer 6. Pracując z dokumentacją udało się uruchomić serwer HTTP który pozwolił na stworzenie strony internetowej(opisanej powyżej) do sterowania i monitorowania parametrów roślin. Na początku adres IP strony był przydzielany automatycznie. Stwierdzono że znacznie łatwiej będzie się posłużyć statycznym adresem IP ustalonym na poziomie kodu. Dzięki temu zawsze strona będzie pod tym samym adresem. Aby nie było konfliktów wybrano adres który nie był używany przez żadne urządzenie w domowej sieci.

Następnie stworzono wygląd strony używając HTML oraz CSS i umieszczono go w kodzie aplikacji. W czasie pracy nad projektem odkryto różne możliwości zapisywania strony oraz stylów w mikokontrolerze. Jedną z możliwości jest zapisanie strony internetowej na karcie pamięci. Inną natomiast jest umieszczenie plików HTML i CSS w systemie plików ESP32. Zamiast wpisywać tekst HTML i CSS do kodu, można utworzyć osobne pliki HTML i CSS, a następnie umieścić je w ESP32 używając SPIFFS (SPI Flash File System). Jest to na pewno ciekawe rozwiązanie, które można również zbadać w przyszłości.

Kolejnym krokiem było wyszukanie odpowiednich czujników do monitorowania parametrów roślin. Zdecydowano się na użycie BMP180 oraz czujników wilgotności gleby. Te czujniki były łatwo dostępne, zostały zastosowane w innych projektach, więc nie było problemu z zamawianiem części. Wyszukano odpowiednie dokumenty opisujące parametry czujników oraz podłączenie. Postępując zgodnie z dokumentacją czujniki zostały podłączone. Użyto również zewnętrznego przetwornika analogowo cyfrowego opisanego w rozdziale 2. Dodatkowo ten element również tak jak BMP180 jest sterowany poprzez magistralę I2C. Dzięki temu zachowano porządek w podłączeniach.

W późniejszym etapie projektu uznano że przydatnym elementem również byłaby możliwość sterowania pompą wody. Pozwalałoby to na zdalne podlewanie rośliny w momencie wykrycia słabej wilgotności. W związku z tym zdecydowano się na dołożenie prostego obwodu sterowania pompą. W najprostszej postaci jest to załączanie i wyłączanie pompy wody poprzez umieszczony na stronie internetowej przycisk ON lub OFF.

Ostatnim elementem było wykorzystanie systemu operacyjnego czasu rzeczywistego FreeRTOS. Jest przeznaczony do systemów czasu rzeczywistego, co oznacza, że może reagować na zdarzenia i sterować procesami w precyzyjnym czasie. Jest to ważne w zastosowaniach, w których synchronizacja ma kluczowe znaczenie, na przykład w systemach sterowania i innych typach systemów czasu rzeczywistego. Dodatkowo FreeRTOS zawiera funkcje zarządzania zasobami systemowymi, takimi jak zadania, kolejki i semaforey, które mogą pomóc w pisaniu wydajnego i niezawodnego oprogramowania. Ważne jest również to, że ma dużą i aktywną społeczność użytkowników i programistów, którzy przyczyniają się do projektu i zapewniają wsparcie i zasoby. Dlatego został wybrany do użycia w tym projekcie. Stworzono zatem odpowiednie taski używając FreeRTOS. Dzięki nim łatwo podzielono pracę procesora na mniejsze fragmenty. Warto zauważyć, że

używane ESP32 jest jednostką dwurdzeniową, co pozwala na lepsze wykorzystanie możliwości FreeRTOS. Głównymi zadaniami procesora w tym projekcie jest obsługa serwera HTTP, oraz dwa zadania polegające na odczycie danych z czujników. Instancja serwera HTTP jest tworzona, przydzielone dla niej zostają pamięć/zasoby w zależności od określonej konfiguracji i wyprowadzony zostaje uchwyt do instancji serwera. Serwer ma zarówno gniazdo nasłuchujące (TCP) dla ruchu HTTP, jak i gniazdo kontrolne (UDP) dla sygnałów sterujących. Priorytet zadania i rozmiar stosu można skonfigurować podczas tworzenia instancji serwera, przekazując strukturę `httpd_config_t` do metody `httpd_start()`. Ruch TCP jest analizowany jako żądania HTTP i w zależności od żądanego URI, wywoływane są zarejestrowane przez użytkownika procedury obsługi, które mają odesłać pakiety odpowiedzi HTTP. W przypadku tego projektu zarejestrowano 3 procedury obsługi. Są to kolejno `uri_get`, `uri_on`, `uri_off`. Pierwsza z nich ma za zadanie wysłać stronę internetową zbudowaną w HTML i CSS oraz uzupełnioną o dane z czujników. Druga pozwala na załączenie pompy wody i wysłanie strony. Trzecia natomiast odpowiada za wyłączenie pompy wody i również wysłanie strony internetowej. Zadania odczytujące dane z czujników zostały podzielone na obsługa czujnika BMP180 i odczyt danych z czujników wilgotności gleby poprzez przetwornik analogowo cyfrowy ADS1115. Dodatkowo zastosowano mutex aby zapobiec jednoczesnemu dostępowi wielu wątków do tego samego zasobu, co może prowadzić do warunków wyścigu i innych problemów. Jest wiele powodów aby zastosować mutexy. Jednym z nich to zsynchronizowanie dostępu do zasobów współdzielonych, takich jak plik lub baza danych. Może to zapobiec konfliktom i uszkodzeniu zasobu. Kolejnym powodem żeby zastosować mutex jest możliwość poprawy wydajności. W niektórych przypadkach użycie muteksu może poprawić wydajność programu wielowątkowego, umożliwiając wielu wątkom jednoczesną pracę nad różnymi zadaniami, zamiast czekać na udostępnienie udostępnionego zasobu.

Kod aplikacji można znaleźć na githubie <https://github.com/JakMir98/PlantMonitoring>

5. Podsumowanie

Założenia zostały zrealizowane. Podczas pracy nad projektem skorzystano z nowego narzędzia - Trello i zapoznano się z nim. Dodatkowo poszerzono swoją wiedzę z zakresu programowania mikrokontrolerów, a w szczególności ESP32. Dzięki temu będzie możliwość realizacji innych projektów w przyszłości lub rozwinięcia tego projektu gdyż ma potencjał. Ostatnim ciekawym aspektem pracy nad tym projektem było rozszerzenie wiedzy na temat FreeRTOS, który umożliwia zarządzanie zadaniami, komunikację między nimi oraz zarządzanie pamięcią, co można wykorzystać do tworzenia skalowalnych aplikacji działających w czasie rzeczywistym.

6. Bibliografia

- [Get Started - ESP32 - — ESP-IDF Programming Guide latest documentation](#)
- [API Reference - ESP32 - — ESP-IDF Programming Guide latest documentation](#)
- [Platformio Docs](#)
- [ADS111x Ultra-Small, Low-Power, I2C-Compatible, 860-SPS, 16-Bit ADCs With Internal Reference, Oscillator, and Programmable Co](#)
- [BMP180 Digital Pressure Sensor](#)
- [FreeRTOS API categories](#)