

PROJEKT

WIZUALIZACJA DANYCH SENSORYCZNYCH

Wizualizacja danych pomiarowych oraz telemetrycznych transmitowanych przez zdalnie stowaną łódź do badania zbiorników wodnych

Antoni Piałucha, 275411



Prowadzący:

dr inż. Bogdan Kreczmer

Katedra Cybernetyki i Robotyki
Wydziału Elektroniki, Fotoniki i
Mikrosystemów

Politechniki Wrocławskiej

4 czerwca 2025

Spis treści

1	Charakterystyka tematu projektu	1
2	Kamienie milowe	1
3	Specyfikacja finalnego produktu	1
4	Terminarz realizacji poszczególnych podcelów	2
5	Diagram Gantta	2
6	Projekt interfejsu graficznego	2
7	Wstępne rezultaty	3
8	Zaawansowane rezultaty.	3
9	Rezultaty prawie końcowe	4
10	Rezultaty końcowe	5

1 Charakterystyka tematu projektu

Projekt dotyczy stworzenia systemu wizualizacji danych zbieranych przez zdalnie sterowaną łódź badawczą. Głównym celem jest opracowanie aplikacji w Qt, która umożliwi prezentację kluczowych parametrów, takich jak lokalizacja GPS, temperatura wody i powietrza oraz inne dane z czujników. Dane będą przesyłane za pomocą protokołu LoRa do mikrokontrolera STM32 Blue Pill, a następnie przekazywane do systemu Linux przez port USB.

Główne cele projektu to:

- Stworzenie wykonywalnego planu pracy i celów projektu.
- Opracowanie aplikacji w Qt do wizualizacji danych w formie wykresów i tabel,
- Stworzenie solidnego systemu komunikacji, odpornego na zakłócenia,

Projekt ten ma na celu ułatwienie monitorowania zbiorników wodnych oraz dostarczenie narzędzia wspomagającego badania środowiskowe.

2 Kamienie milowe

Aby osiągnąć założone cele, projekt został podzielony na kilka etapów realizacji:

- Przegląd literatury i zasobów internetowych dotyczących technologii LoRa, STM32 oraz Qt.
- Opracowanie założeń i harmonogramu projektu.
- Opracowanie wyglądu interfejsu użytkownika.
- Zaimplementowanie prezentacji logów w wygodnej dla użytkownika formie.
- Zaimplementowanie prezentacji lokalizacji łodzi w wygodnej dla użytkownika formie.
- Zaimplementowanie prezentacji wartości mierzonych w wygodnej dla użytkownika formie.
- Testowanie całego systemu, optymalizacja wydajności oraz odporności na błędy transmisji,
- Dokumentacja techniczna oraz opracowanie raportu końcowego.

3 Specyfikacja finalnego produktu

Finalnym efektem projektu będzie system składający się z:

- Zdalnie sterowanej łodzi wyposażonej w czujniki pomiarowe (temperatura, GPS, itp.),
- Modułu komunikacji LoRa do przesyłu danych,
- Oprogramowania na STM32 do obsługi czujników i transmisji,

- Aplikacji w Qt działającej w systemie Linux, umożliwiającej wizualizację danych w czasie rzeczywistym oraz analizę zapisanych logów,
- Dokumentacji opisującej implementację systemu oraz jego możliwości.

Zakładana dokładność pomiarów:

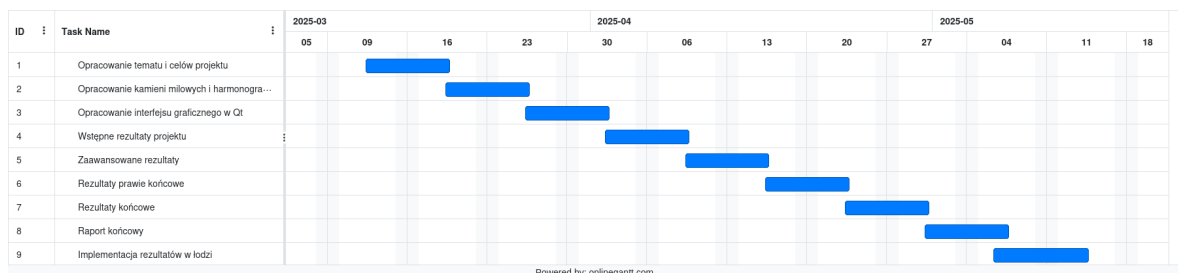
- Lokalizacja GPS: dokładność do 2-5 metrów,
- Pomiar temperatury: dokładność do $\pm 0.5^{\circ}\text{C}$,
- Czas przesyłu danych: poniżej 5 sekund w normalnych warunkach transmisji.

4 Terminarz realizacji poszczególnych podcelów

Harmonogram prac został zaplanowany w następujący sposób:

- 19 marca 2025 – Opracowanie tematu i celów projektu
- 26 marca 2025 – Opracowanie kamieni milowych i harmonogramu projektu.
- 2 kwietnia 2025 – Opracowanie interfejsu graficznego w Qt.
- 9 kwietnia 2025 – Wstępne rezultaty projektu.
- 16 kwietnia 2025 – Zaawansowane rezultaty.
- 23 kwietnia 2025 – Rezultaty prawie końcowe.
- 30 kwietnia 2025 – Rezultaty końcowe.
- 7 maja 2025 – Raport końcowy.
- 14 maja 2025 – Implementacja rezultatów w łodzi.

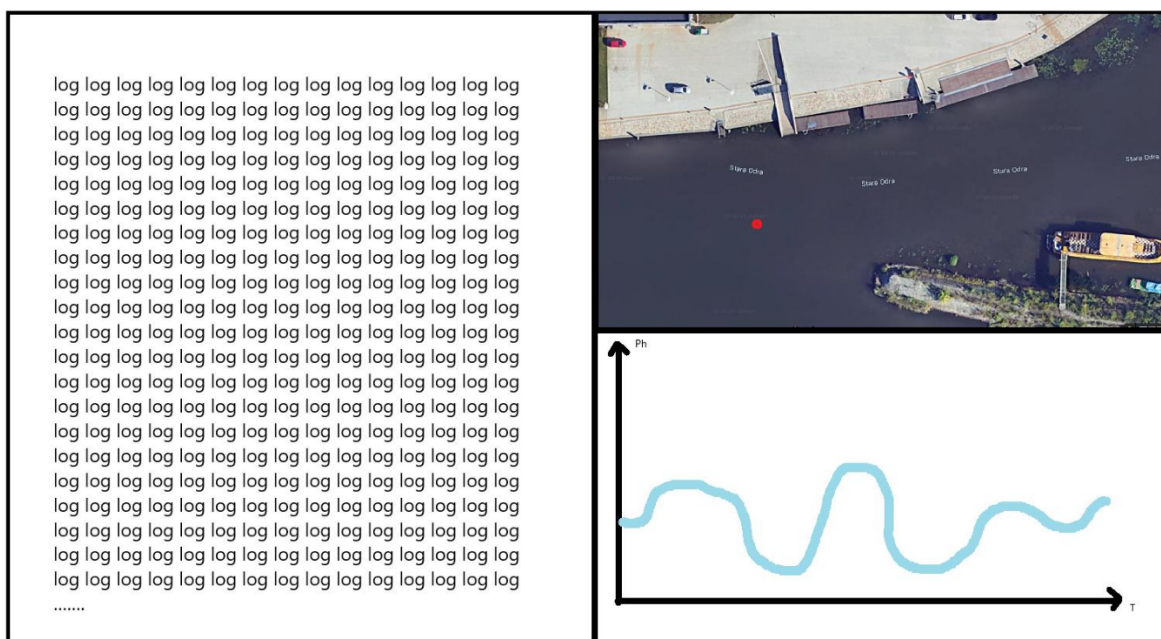
5 Diagram Gantta



Rysunek 1: Diagram Gantta

6 Projekt interfejsu graficznego

Wizja interfejsu graficznego znajduje się na obrazku (2).



Rysunek 2: Wizja interfejsu graficznego.

7 Wstępne rezultaty

Na obecnym etapie przygotowano prowizoryczny interfejs użytkownika z wykorzystaniem układów *layout* w środowisku Qt. Aplikacja zawiera:

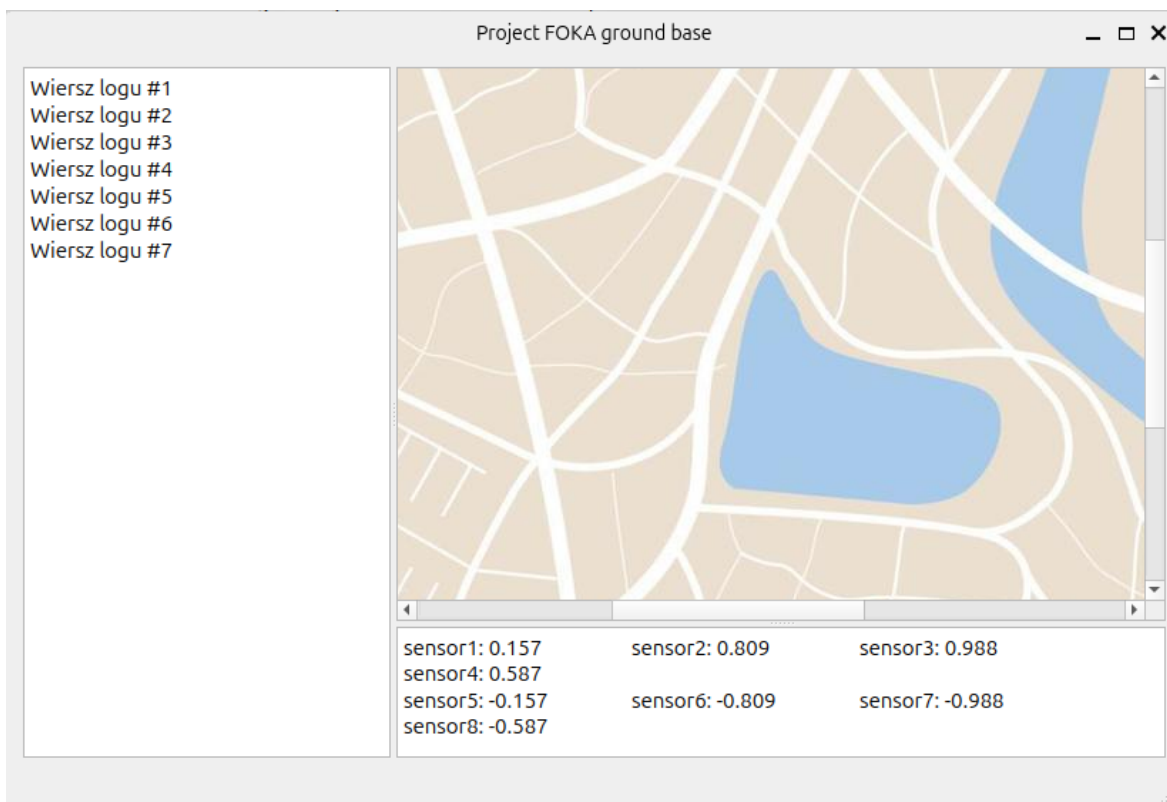
- generowanie przykładowych wpisów w logu, które pojawiają się cyklicznie w konsoli tekstowej,
- symulowane dane z czujników, aktualizowane co sekundę i wyświetlane w czytelnej formie,
- podgląd mapy – w rogu interfejsu osadzono obraz mapy w postaci statycznego zdjęcia.

Dodatkowo, do projektu została dodana wstępna dokumentacja techniczna generowana za pomocą narzędzia Doxygen. Protokół komunikacji będzie opierał się na przesyłaniu szeregu danych z czujników w ustalonej kolejności. Następnie zostaną wysłane logi. Dane będą przedzielone średnikami, a paczka będzie zawsze kończyła się znakiem nowej linii. Np. 0.111;256;up;9.11;Queue Closed \n.

Zrzut ekranu przedstawiający aktualny wygląd interfejsu znajduje się na rysunku 3.

8 Zaawansowane rezultaty.

Układ graficzny aplikacji został częściowo zmodyfikowany względem pierwotnej koncepcji. Logi systemowe wciąż wyświetlane są w lewym górnym rogu okna, a mapa pozostaje po stronie prawej. W prawym dolnym rogu umieszczono teraz bieżące wartości odczytów z czujników. Dane pomiarowe prezentowane na wykresach będą natomiast wyświetlane w osobnym, nowym oknie.



Rysunek 3: Wstępne rezultaty.

Do dnia 16.05.2025 aplikacja została rozbudowana o funkcjonalność odbierania danych z portu szeregowego, zgodnie z wcześniej zdefiniowanym protokołem transmisji. Program analizuje otrzymane pakiety, przetwarza zawarte w nich informacje i rozmieszcza je w odpowiednich częściach interfejsu użytkownika.

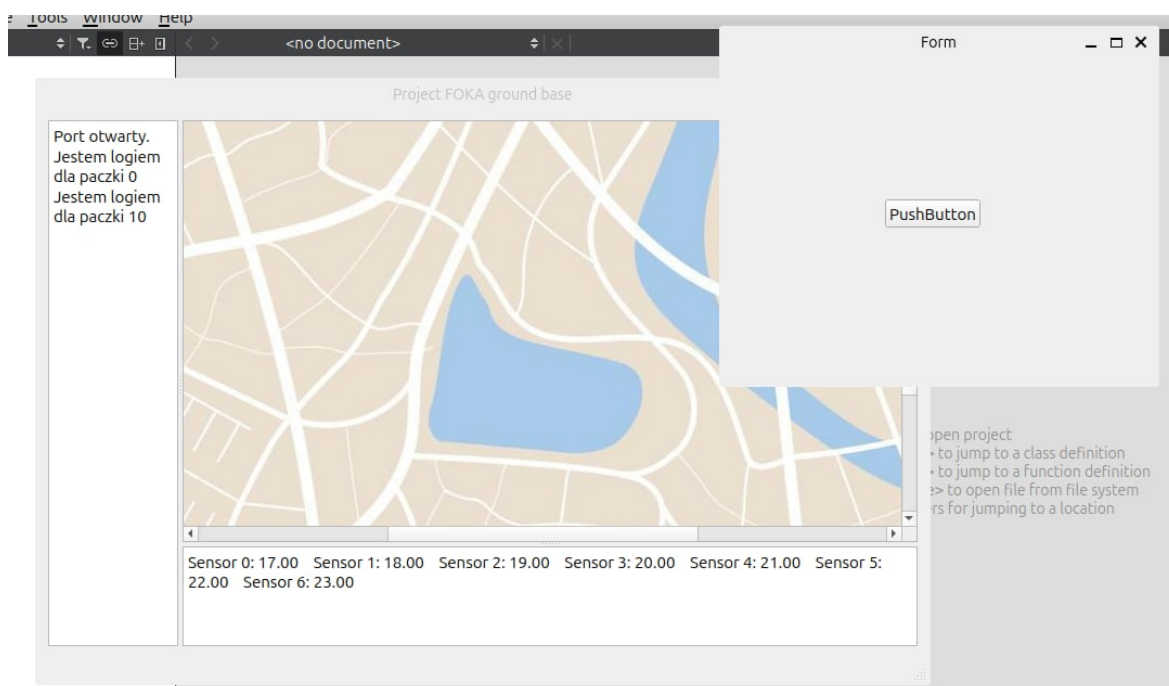
Dodatkowo utworzono nową klasę odpowiedzialną za prezentację wykresów w oddzielnym oknie. Chociaż komponent ten jest już zintegrowany z resztą programu, wykresy nie są jeszcze wyświetlane. Aktualny wygląd interfejsu przedstawiono na rysunku 4.

9 Rezultaty prawie końcowe

W ramach dalszego rozwoju aplikacji graficznej Project FOKA zrealizowano pełną obsługę komunikacji szeregowej oraz zintegrowano interfejs użytkownika z systemem akwizycji danych. Program odbiera pakiety danych zgodne z ustalonym protokołem i poprawnie je interpretuje.

Odczytane informacje trafiają do różnych komponentów interfejsu: dane lokalizacyjne aktualizują pozycję łodzi na mapie, wartości sensorów wyświetlane są w dolnej części głównego okna w postaci kafelków, a logi systemowe trafiają do konsoli po lewej stronie. Dodatkowo, sygnał pochodzący z czujnika głębokości automatycznie zmienia kolor tła kafelka na czerwony w przypadku przekroczenia głębokości krytycznej.

Warto zaznaczyć, że wszystkie dane numeryczne są również przekazywane do osobnego okna z wykresami. Każdy czujnik prezentowany jest na osobnym wykresie liniowym, przy czym dla każdego z nich można określić jednostkę, zakres wartości oraz kolor krzywej na podstawie pliku konfiguracyjnego JSON.



Rysunek 4: Zaawansowane rezultaty.

Oprogramowanie umożliwia również dynamiczną zmianę języka interfejsu graficznego (PL/EN), z użyciem dedykowanych przycisków graficznych znajdujących się w prawym górnym rogu obszaru mapy. Elementy interfejsu tłumaczone są automatycznie na podstawie załadowanych plików `.qm`.

Całość projektu została w pełni udokumentowana przy użyciu systemu Doxygen. Każdy plik źródłowy oraz nagłówkowy zawiera opisy klas, metod, sygnałów i struktur danych. Wygenerowaną dokumentację można znaleźć w folderze `docs/html` w formacie przeglądarkowym. Projekt nie wymaga już dalszych znacznych poprawek.

Aktualny stan działania interfejsu przedstawiono na rysunku 5.

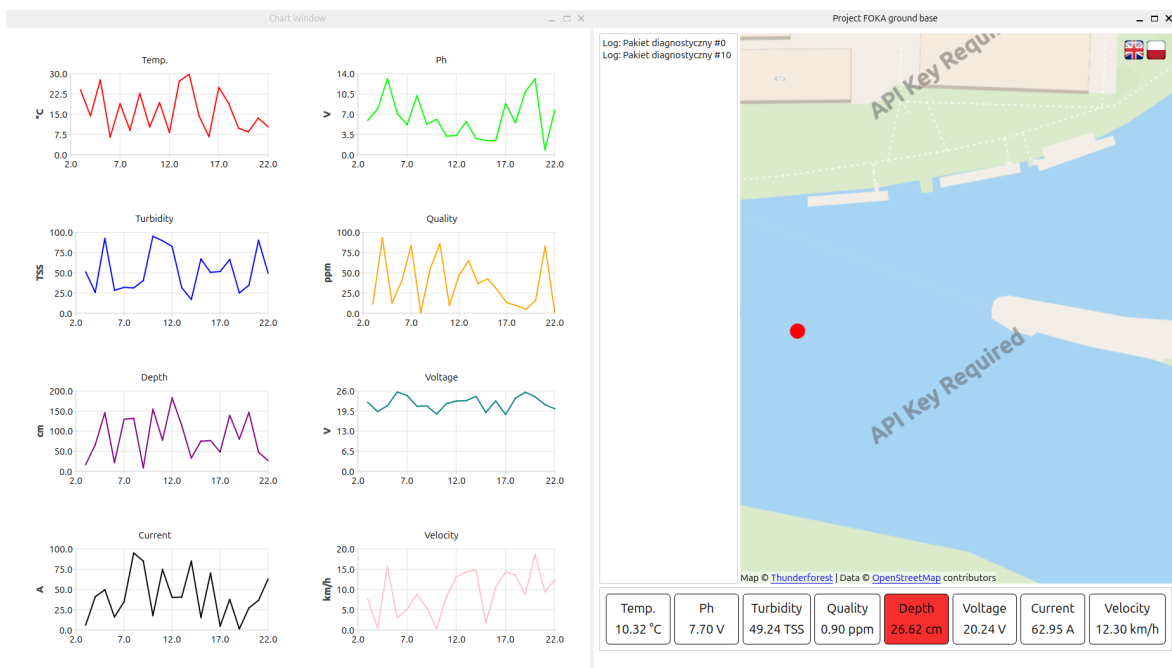
10 Rezultaty końcowe

W finalnej wersji aplikacji nie wprowadzono znaczących zmian w zakresie funkcjonalności ani struktury programu w porównaniu do wersji opisanej w poprzedniej sekcji. Interfejs użytkownika, system komunikacji szeregowej oraz sposób prezentacji danych pozostały bez zmian i działają zgodnie z założeniami projektowymi.

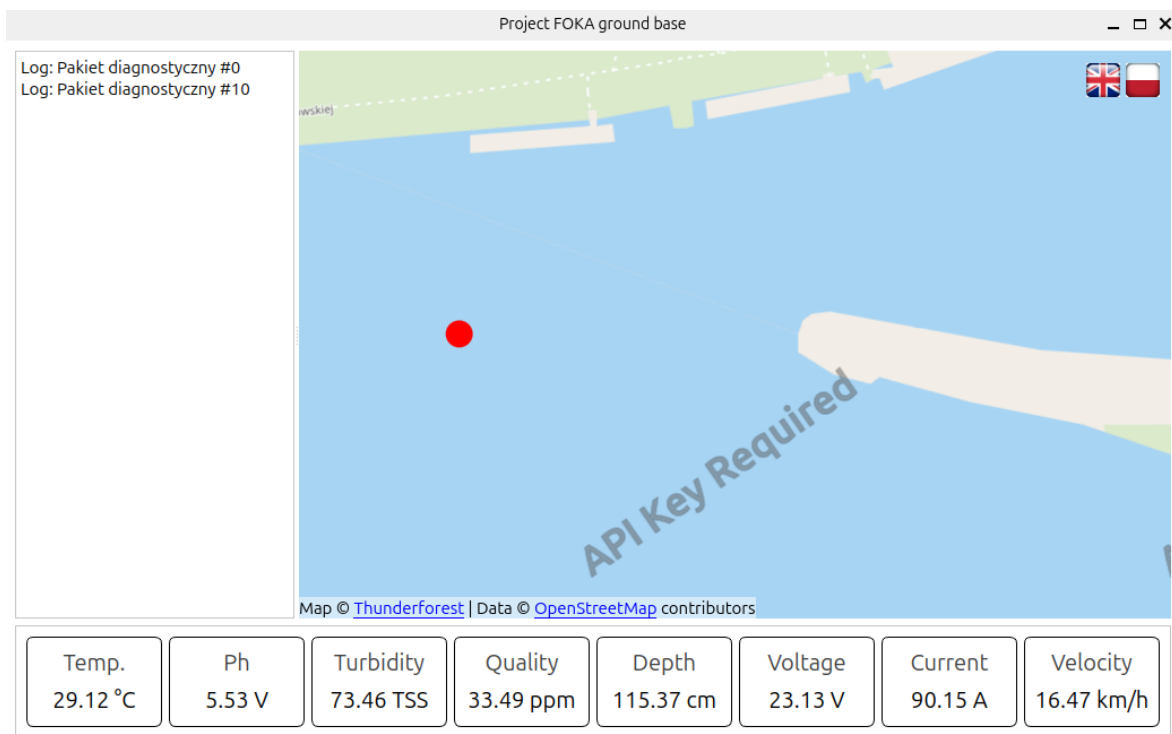
Wprowadzono jedynie drobne poprawki estetyczne w głównym oknie aplikacji. Zmieniono minimalną szerokość etykiet (labeli) wyświetlających dane z sensorów, dzięki czemu ich rozmiar pozostaje stały niezależnie od zmieniających się nazw czujników lub wartości pomiarowych. Zapobiega to niepożądanym przesunięciom układu graficznego podczas pracy aplikacji.

Dodatkowo, nazwy sensorów w etykietach zostały wizualnie wyróżnione poprzez nadanie im lekko szarawego koloru. Zmiana ta poprawia czytelność interfejsu, pozwalając użytkownikowi w prostszy sposób skupić wzrok na aktualnych wartościach pomiarowych, jednocześnie zachowując przejrzystość układu informacji.

Finalny wygląd interfejsu przedstawiono na rysunku 6.



Rysunek 5: Interfejs aplikacji z aktywnym systemem komunikacji i wizualizacji.



Rysunek 6: Finalny wygląd okna głównego aplikacji.