

# PROJEKT

WIZUALIZACJA DANYCH SENSORYCZNYCH

---

## Stacja pomiarowa do kontroli stanu wody w zbiorniku

---

Paweł Łyszczarz, 259258

---

---

---



*Prowadzący:*  
dr inż. Bogdan Kreczmer

Katedra Cybernetyki i Robotyki  
Wydziału Elektroniki, Fotoniki i  
Mikrosystemów  
Politechniki Wrocławskiej

7 października 2023

# Spis treści

<b>1</b>	<b>Charakterystyka tematu projektu</b>	<b>1</b>
<b>2</b>	<b>Podcele i etapy realizacji projektu</b>	<b>1</b>
<b>3</b>	<b>Specyfikacja finalnego produktu</b>	<b>1</b>
<b>4</b>	<b>Terminarz realizacji poszczególnych podcelów (z dokładnością do 1 tygodnia)</b>	<b>1</b>
<b>5</b>	<b>Projekt graficznego interfejsu użytkownika</b>	<b>4</b>
5.1	Funkcjonalności aplikacji . . . . .	4
5.2	Wizualizacja projektu . . . . .	4
5.2.1	Okno widoku bazowego - ogólnego . . . . .	4
5.2.2	Okno podglądu wykresów . . . . .	5
<b>6</b>	<b>Wstępne rezultaty</b>	<b>6</b>
6.1	Opis rezultatów . . . . .	6
6.2	Protokół komunikacyjny UART . . . . .	8
<b>7</b>	<b>Rezultaty zaawansowane</b>	<b>9</b>
7.1	Wygląd aplikacji . . . . .	9
7.2	Dane z układu rzeczywistego . . . . .	11
<b>8</b>	<b>Rezultaty prawie końcowe</b>	<b>12</b>
8.1	Wygląd aplikacji . . . . .	12
<b>9</b>	<b>Rezultaty końcowe</b>	<b>13</b>
9.1	Funkcjonalność aplikacji . . . . .	13
9.2	Aspekt wyglądu aplikacji . . . . .	13
9.2.1	Okno główne . . . . .	13
9.2.2	Okno wykresów . . . . .	15
9.3	Komunikacja przez port szeregowy . . . . .	16
9.4	Wady aplikacji . . . . .	16

# 1 Charakterystyka tematu projektu

Zamysłem projektu jest stworzenie i wizualizacja danych sensorycznych ze stacji pomiarowej na zbiorniku wodnym, wyposażonej w między innymi w czujnik wilgotności oraz temperatury, wysokości wody jak zarówno prędkość przepływu wody w symulowanym "kanale". Projekt od strony fizycznej zostanie skonstruowany z użyciem płytki Arduino Uno. Od strony wizualizacji przewiduję, że projekt będzie ujawniał graficzne odzwierciedlenie stanu zbiornika i wszystkich informacji z nim związanych.

## 2 Podcele i etapy realizacji projektu

Planowane są poszczególne etapy, lista podcelów:

- Przegląd literatury i zasobów Internetu związanych z tematem projektu
- Projekt układu elektronicznego (schemat ideowy)
- Stworzenie całego układu elektronicznego.
- Zaprojektowanie wizualizacji danych sensorycznych.
- Implementacja założeń projektowych.
- Dopracowanie komunikacji układu z komputerem.
- Testy aplikacji oraz układu i ewentualna redukcja błędów.

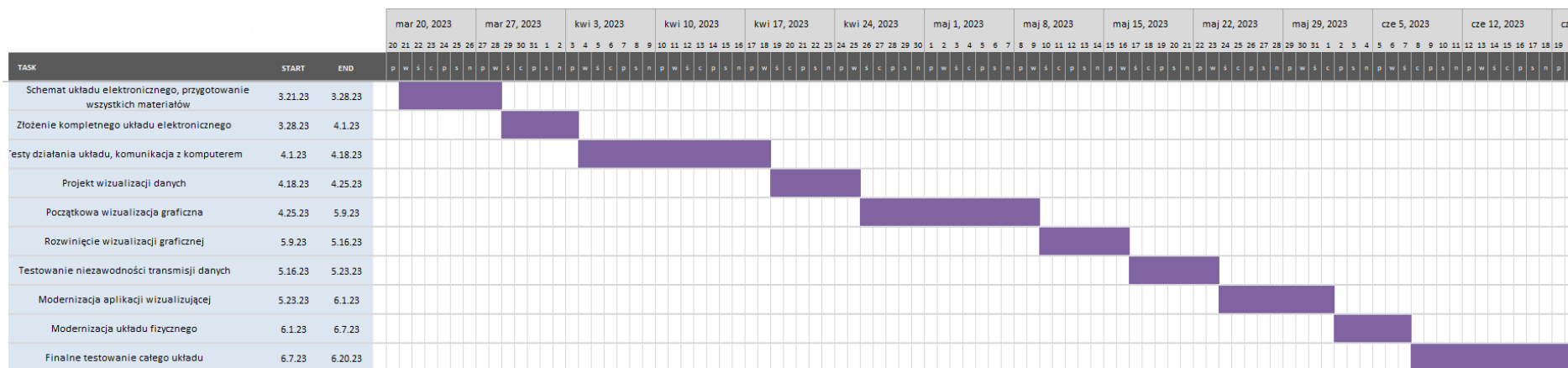
## 3 Specyfikacja finalnego produktu

Projekt na finalnym etapie będzie zawierał aplikację wizualizującą dane ze stacji pomiarowej kontrolującej stan zbiornika, którego temperaturę, poziom wody oraz prędkość przepływu będzie można kontrolować, a aplikacja odpowiednio te zmiany uwzględniać. Projekt takiego zbiornika ma odzwierciedlenie na przykład w stacjach anty-powodziowych, gdzie stan wysokości wody oraz prędkości przypływów są stale monitorowane.

## 4 Terminarz realizacji poszczególnych podcelów (z dokładnością do 1 tygodnia)

- 21 marca 2023 – zakończenie przeglądu materiałów związanych z danym tematem, podjęcie decyzji dotyczącej wyboru czujników oraz ich zakup.
- 28 marca 2023 – projekt aplikacji do wizualizacji danych, zaplanowanie wyglądu odpowiednich elementów interfejsu oraz wstępne zapoznanie się z metodami ich implementacji.
- 4 kwietnia 2023 – schemat układu elektronicznego, przygotowanie wszystkich materiałów, początkowy projekt stanowiska pomiarowego, symulacyjny zbiornik na wodę z miejscami na czujniki oraz „kanał” do pomiaru prędkości przepływu.

- 18 kwietnia 2023 – złożenie kompletnego układu elektronicznego, podłączenie wszystkich czujników do kontrolera, umieszczenie ich w ostatecznych miejscach pomiarowych oraz zapoznanie się z ich działaniem.
- 25 kwietnia 2023 – początkowa wizualizacja graficzna, wdrożenie poznanych metod przy założeniach projektowych.
- 9 maja 2023 – testy działania układu, komunikacja z komputerem, próby znalezienia błędów, testowanie czujników, manipulowanie stanem zbiornika wodnego.
- 16 maja 2023 – rozwinięcie wizualizacji graficznej oraz wcześniej utworzonych elementów graficznych przedstawiających dane z czujników.
- 23 maja 2023 – testowanie niezawodności transmisji danych, testowanie limitów czujników i reakcji aplikacji na te limity.
- 1 czerwca 2023 – Dopracowanie aplikacji wizualizującej oraz interfejsu i jego elementów.
- 7 czerwca 2023 – Dopracowanie układu fizycznego, stworzenie łatwego do testów środowiska.
- 20 czerwca 2023 – finalne testowanie całego układu



## 5 Projekt graficznego interfejsu użytkownika

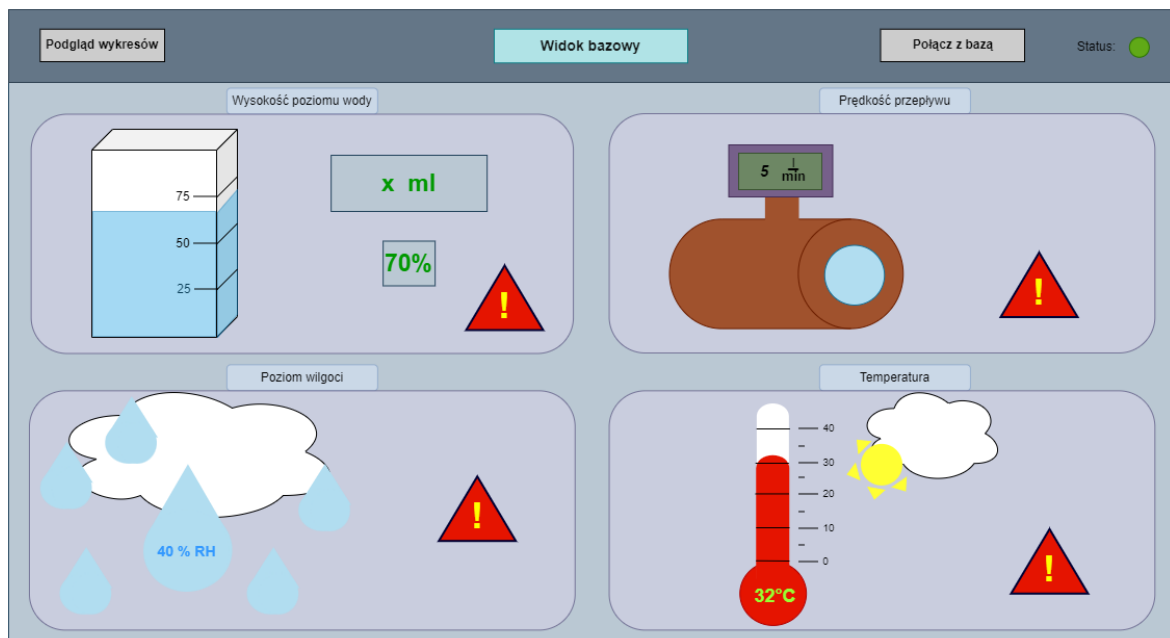
### 5.1 Funkcjonalności aplikacji

Aplikacja będzie dostarczać interfejs użytkownika ukazujący widok ogólny (bazowy) stanu wszystkich czujników na raz. W tym ognie widoczny będzie przycisk połączenia komunikacji aplikacji ze stacją pomiarową oraz lampkę LED sygnalizującą o statusie połączenia.

### 5.2 Wizualizacja projektu

#### 5.2.1 Okno widoku bazowego - ogólnego

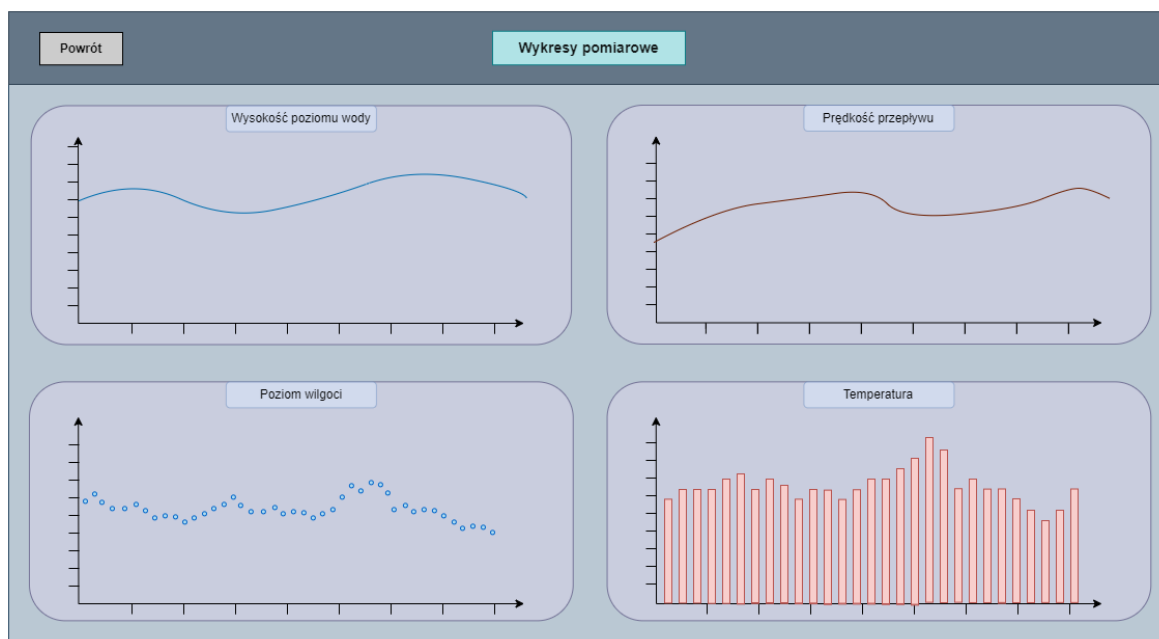
Po uruchomieniu aplikacji ukaże się widok bazowy, który pozwalać będzie na kontrolę wszystkich odczytów z czujników na raz 2. Każdy z pomniejszych prostokątów będzie przedstawiał odczytywane z konkretnych czujników dane. W zależności od tego jak blisko górnej granicy zakresu pomiarowego czujnika, mierzony obiekt będzie się znajdował istnieje możliwość pojawienia się czerwonego trójkąta z żółtym wykrzyknikiem w środku w celu zawiadomienia i ostrzeżenia użytkownika aplikacji o tych zdarzeniach. W lewym górnym rogu widnieć będzie również przycisk pozwalający na podgląd wykresów odzwierciedlających przebieg pomiarów każdego z sensorów z osobna w przestrzeni czasu w nowym oknie 3.



Rysunek 2: Widok bazy

## 5.2.2 Okno podglądu wykresów

Po wciśnięciu przycisku "Podgląd wykresów" z pozycji widoku bazowego otworzy się nowe okno podobnie wyglądające do poprzedniego, z tą różnicą, że teraz w prostokątach wcześniej ujawniających modele wizualizujące stan mierzonych obiektów znajdować się będą wykresy dokumentujące przebieg pomiarów czujników w przestrzeni czasu. W lewym górnym rogu widoczny będzie przycisk pozwalający na powrót do standardowego widoku bazowego.

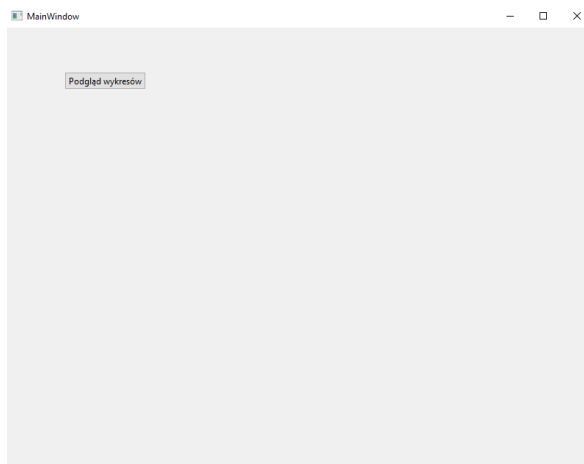


Rysunek 3: Widok wykresów

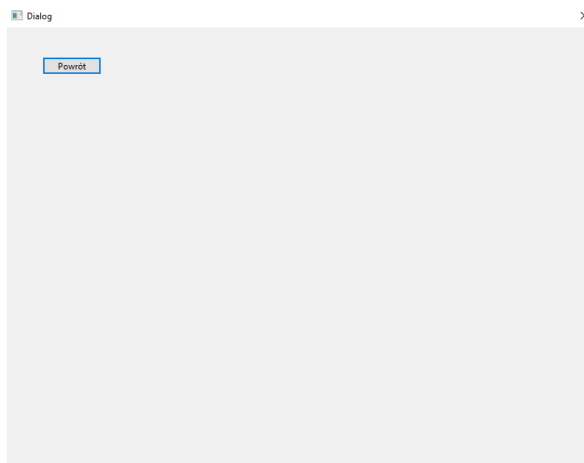
## 6 Wstępne rezultaty

### 6.1 Opis rezultatów

W ramach zaplanowanych działań w harmonogramie, do tego momentu został złożony kompletny schemat elektroniczny, zawierający czujniki oraz płytkę z mikroprocesorem, udało się również pomyślnie skomunikować stację pomiarową z komputerem poprzez interfejs komunikacyjny UART oraz poprawnie odebrać te przesyłane dane w QT Application w utworzonym projekcie, poprzez prawidłowe zapisywanie osobnych odczytów do tablicy. Udało się również skonstruować dwa docelowe okna, jedno w przyszłości przedstawiające widok bazowy aplikacji ze wszystkimi pomiarami z czujników 4, oraz okno z podglądem na wykresy pomiarowe 5. Obydwa z tych ekranów posiadają przyciski pozwalające na przełączanie się między ekranami oraz ich ukrywanie.



Rysunek 4: Okno bazowe



Rysunek 5: Okno wykresów



Kod do obsługi przełączania się pomiędzy okienkami za pomocą przycisków

1. main\_window

```
QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    MainWindow(QWidget *parent = nullptr);
    ~MainWindow();

private slots:
    void on_pushButton_clicked();

private:
    Ui::MainWindow *ui;
    Second_Dialog *sec_dial;
};
```

Rysunek 6: *main\_window.h*

```
MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    ui->setupUi(this);
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::on_pushButton_clicked()
{
    hide();
    sec_dial = new Second_Dialog(this);
    sec_dial->show();
}
```

Rysunek 7: *main\_window.cpp*

## 2. second\_dialog

```
class MainWindow;

namespace Ui {
class Second_Dialog;
}

class Second_Dialog : public QDialog
{
    Q_OBJECT

public:
    explicit Second_Dialog(MainWindow *main_window, QWidget *parent = nullptr);
    ~Second_Dialog();

private slots:
    void on_pushButton_clicked();
    void on_backButton_clicked();

private:
    Ui::Second_Dialog *ui;
    MainWindow *main_window;
};
```

Rysunek 8: *second\_dialog.h*

```
Second_Dialog::Second_Dialog(MainWindow *main_window, QWidget *parent) :
    QDialog(parent),
    ui(new Ui::Second_Dialog),
    main_window(main_window)
{
    ui->setupUi(this);

    connect(ui->pushButton, &QPushButton::clicked, this, &Second_Dialog::on_pushButton_clicked);
}

Second_Dialog::~Second_Dialog()
{
    delete ui;
}

void Second_Dialog::on_pushButton_clicked()
{
    main_window->show();
    close();
}
```

Rysunek 9: *second\_dialog.cpp*

## 6.2 Protokół komunikacyjny UART

W ramach komunikacji stacji mierniczej z urządzeniem został skonfigurowany protokół komunikacyjny UART. Mikroprocesor po porcie szeregowym wysyła odpowiednio przetworzone dane z czujników w postaci linii tekstu, w którym każdy pomiar jest oddzielony przecinkiem, dodatkowo do każdej linijki przesyłanej portem szeregowym po przecinku i znaku mnożenia (,) zostaje umieszczona suma kontrolna będąca sumą wszystkich wartości przesyłanych danych zapisanych w systemie sześciennym 10. Po odpowiednim przetworzeniu otrzymanych danych przez Qt Application oraz obliczeniu i sprawdzeniu sumy kontrolnej, dane są zapisywane do buffora 11.



Rysunek 10: Przesyłany pasek danych

```

Found serial port: "COM3"
Received data: 22 49 5 0 "4C"
Received data: 22 49 4 0 "4B"
Received data: 22 49 4 0 "4B"
Received data: 22 50 6 0 "4E"
Received data: 22 52 4 0 "4E"
Received data: 22 58 6 0 "56"
Received data: 22 64 5 0 "5B"

```

Rysunek 11: Odbierane dane

## 7 Rezultaty zaawansowane

Do tego momentu udało się zrealizować wszystkie dotychczas zaplanowane cele i etapy realizacji projektu, od poprawnie przesyłanych danych do początkowego wyglądu aplikacji

### 7.1 Wygląd aplikacji

Wygląd odpowiednich okien aplikacji odpowiadających poszczególnym widokom danych zostały dostosowane na wzór wcześniej zaprojektowanych wyglądom widoków 12.



Rysunek 12: Okno widoku bazy

Okno główne ukazujące widok bazy zawiera cztery główne pola z których każde jest opisane ramką nazwy np. "Wysokość poziomu wody", są one równo rozłożone na ekranie a ich rozmiar odpowiednio dostosowuje się to wielkości okna aplikacji. Każde wyżej wymienione pole posiada w sobie obiekty, które w dalszych etapach będą odpowiadały za odpowiednie grafiki i wyświetlane dane. Położenie tych obiektów jest ściśle określone w danych polach i nie zmienia się ono pomimo skalowania okna. Na ramce powyżej

wspomnianych pól widnieje nazwa obecnie wyświetlanego widoku oraz funkcjonalne przyciski takie jak Podgląd wykresów który otwiera nowe okno z widokiem wykresów, czy też przycisk do połączenia z bazą wraz z diodą LED wskazującej na stan połączenia.



Rysunek 13: Okno widoku wykresów

Drugie okno - okno widoku wykresów posiada identycznie rozmieszczone i nazwane dynamicznie skalujące się pola, w których tym razem znajdować się będą wykresy dokumentujące przebieg wartości danych, które tak samo będą dostosowywać swój rozmiar i położenie do otoczenia 13. Podobnie jak poprzednio nad wspomnianymi polami znajduje się ramka z nazwą aktualnie wyświetlanego widoku oraz przycisk powrotu zamykający okno z wykresami.

## 7.2 Dane z układu rzeczywistego

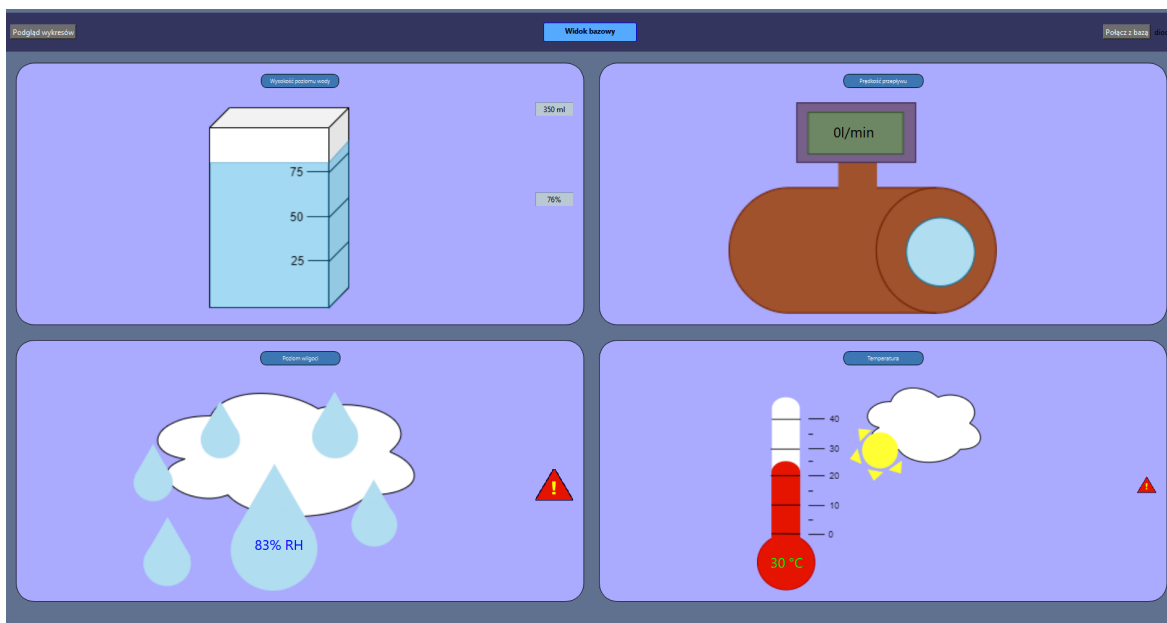
Układ oraz dane z niego otrzymywane w aplikacji zostały sprawdzone pod względem poprawności i trafności odczytów pomiarowych. Odpowiednio odbierane, sprawdzane i filtrowane ramki danych są przechowywane w programie i gotowe do wyświetlania na ich bazie grafik ilustrujących stan bazy oraz przebieg odczytywanych danych.

## 8 Rezultaty prawie końcowe

W tym etapie udało się zrealizować wszystkie dotychczas zaplanowane postępy. Okno główne jest już praktycznie całkowicie ukończone.

### 8.1 Wygląd aplikacji

Główne okno aplikacji jest już wyposażone w obiekty wizualizujące odczyty wszystkich czujników, wraz z wyświetleniem ich dosłownych wartości 14.



Rysunek 14: Okno główne

Dodane również zostały znaki ostrzegawcze, przy wcześniej ustalonych granicach bezpieczeństwa parametrów odczytywanych przez sensory, wyświetlany jest znak który sygnalizuje zagrożenie lub stan wymagający szczególnej uwagi.

## 9 Rezultaty końcowe

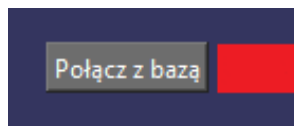
Projekt został zakończony pomyślnie, wszystkie założone etapy zostały zrealizowane w terminie. Utworzono specjalny plik wykonywalny (eng. executable file ".exe"), który pozwala na uruchomienie aplikacji na każdym komputerze bez względu na wersję aplikacji Qt.

### 9.1 Funkcjonalność aplikacji

Aplikacja dostarcza główny widok okna bazowego z interfejsem użytkownika, który pozwala na interakcję z bazą czy też na podgląd jej dodatkowych funkcjonalności. Oprócz głównego widoku aplikacja posiada możliwość przełączenia się na drugie okno, w którym mamy dostęp do na bieżąco wyrysowywujących się wykresów.

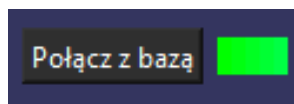
### 9.2 Aspekt wyglądu aplikacji

Po początkowym uruchomieniu aplikacji ukazuje nam się okno główne aplikacji nieposiadające jeszcze jednak żadnej grafiki odzwierciedlającej odczyty z czujników, w zasadzie to i dane z portu szeregowego nie są jeszcze na ten moment odczytywane. Opisywany stan można rozpoznać poprzez czerwoną ikonę znajdującą się w prawym górnym rogu aplikacji, tuż przy przycisku "Połącz z bazą" 15



Rysunek 15: Sygnalizator niepołączonej stacji mierniczej

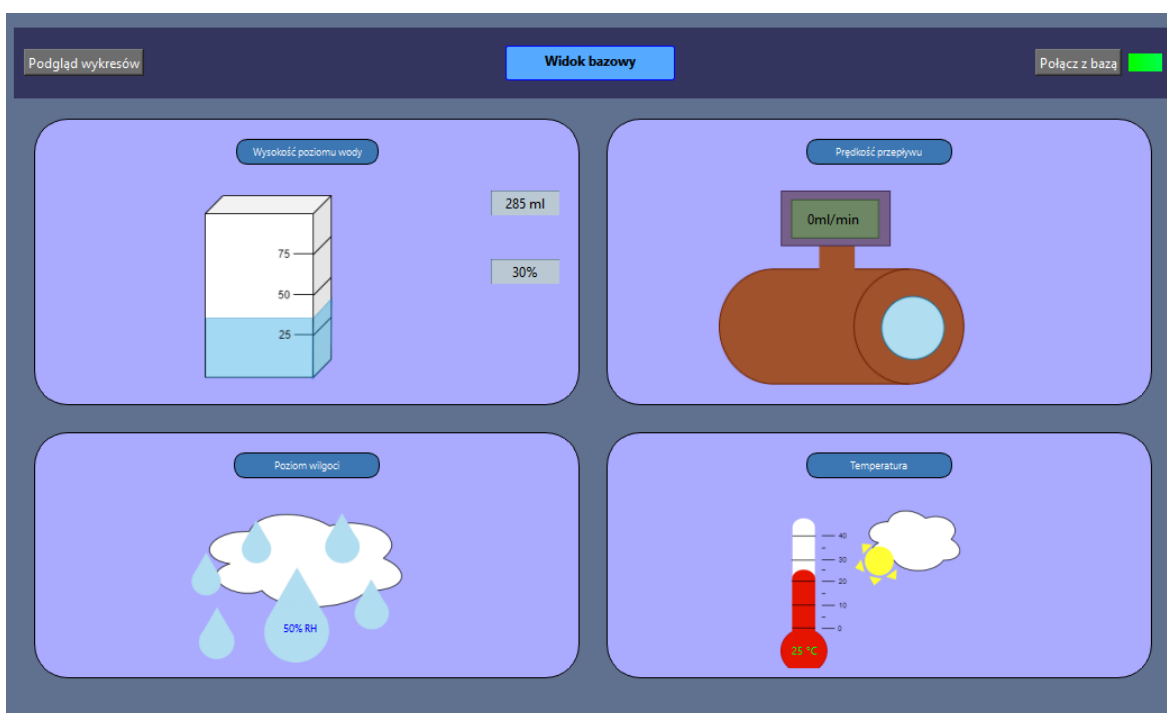
Po wciśnięciu przycisku "Połącz z bazą" aplikacja powinna automatycznie znaleźć dostępny port szeregowy a następnie wyświetlić docelowe grafiki wizualizujące odczyty z czujników oraz odpowiednio zasygnalizować poprawne połączenie z bazą poprzez zieloną ikonę znajdującą się również w prawym górnym rogu 16



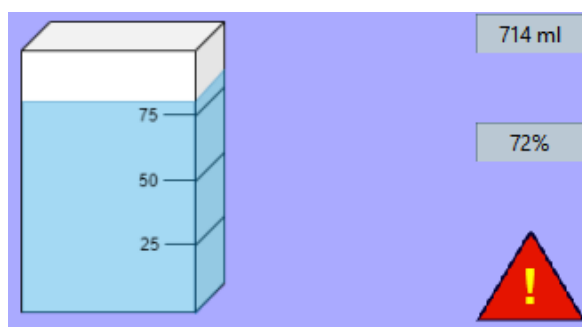
Rysunek 16: Sygnalizator połączonej stacji mierniczej

#### 9.2.1 Okno główne

Przy poprawnie połączonej stacji mierniczej powinien nam się pokazać okno główne w pełnej okazałości ze wszystkimi grafikami 17. W każdym z zaokrąglonych czworokątów widoczne są osobne grafiki odpowiadające każdemu z czujników w układzie rzeczywistym. W lewym górnym zaokrąglonym czworokącie widoczny jest zbiornik wody, którego poziom zmienia się w zależności od poziomu wody w rzeczywistym pojemniku, ponadto obok zbiornika znajdują się napisy, górny z dokładną ilością wody w mililitrach a niżej procentowe wypełnienie pojemnika, w przypadku przekroczenia ustalonej granicy wysokości wody, obok napisów pojawi się czerwony trójkąt z żółtym wykrzyknikiem oznaczający ostrzeżenie przed zagrażającym poziomem wody 18.



Rysunek 17: Okno główne aplikacji



Rysunek 18: Ostrzeżenie o wysokości wody

W prawym górnym rogu natomiast widoczna jest grafika odzwierciedlająca odczyty z czujnika prędkości przepływu wody. Na zielonym panelu wyświetlana jest dokładna wartość odczytywana przez czujnik w jednostce mililitrów na minuty. W przypadku przekroczenia wcześniej ustalonej granicy również wyświetlany jest trójkąt ostrzegawczy 19.

Analogicznie sytuacja wygląda w prawym dolnym i lewym dolnym zaokrąglonym czworokącie, gdzie zarówno dla wizualizowanego czujnika poziomu wilgoci jak i dla czujnika temperatury, po przekroczeniu granicy bezpieczeństwa przez odczyty z czujników zaświeca się trójkąt ostrzegawczy 20. Po spadku wartości odczytu z czujnika poniżej wartości granicznej, ostrzeżenie w postaci trójkąta naturalnie znika.





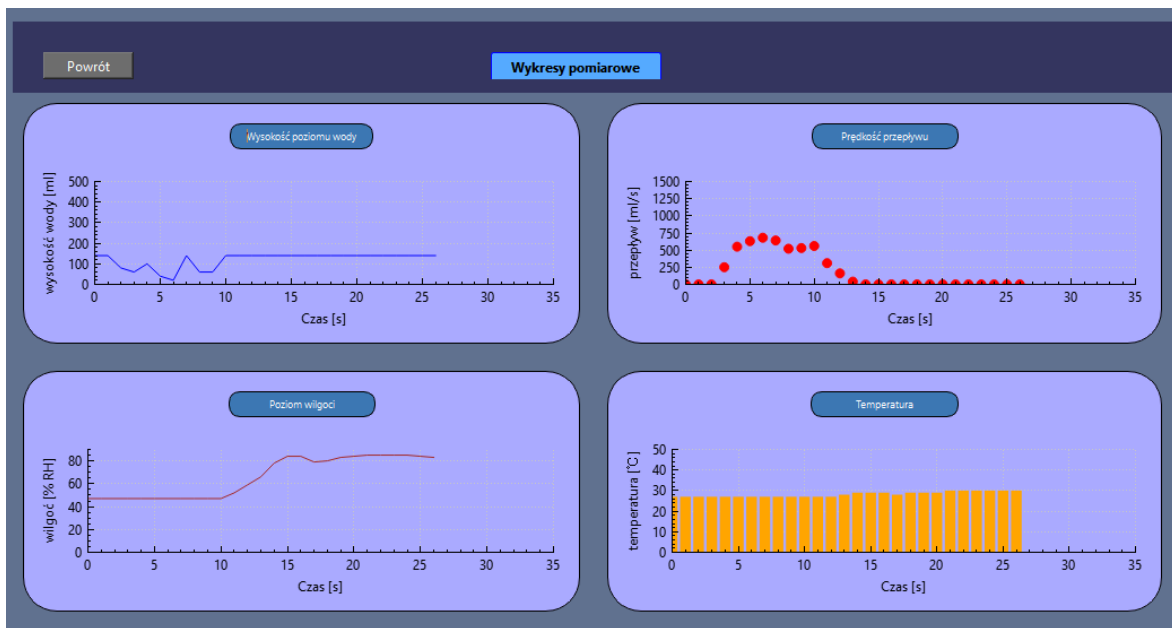
Rysunek 19: Ostrzeżenie o prędkości przepływu wody



Rysunek 20: Ostrzeżenie o poziomie wilgoci

### 9.2.2 Okno wykresów

W oknie wykresów zostały stworzone identyczne zaokrąglone czworokąty, tym razem zawierające wykresy przedstawiające pomiary każdego z czujników w zależności od czasu 21. Prawie dla każdego z wykresów zastosowany inny rozmiar plotowania aby pokazać możliwości zawartej biblioteki do tworzenia wykresów.



Rysunek 21: Okno wykresów

### 9.3 Komunikacja przez port szeregowy

Od poprzedniego momentu implementacji komunikacji aplikacji z portem szeregowym nic nie zostało zmienione. Przetestowana została tylko niezawodność transmisji danych z wynikiem pozytywnym. Ramka wysyłanych danych dalej ma format jak poprzednio 10, a przerobione i podzielone już dane przez aplikację prezentują się bez zmian tak jak wcześniej 11.

### 9.4 Wady aplikacji

Jedyną zauważalną lecz dopuszczalną wadą aplikacji jest odczyt poziomu wody. W założeniu układ rzeczywisty miał składać się z bardziej wysokiego niż szerokiego pojemnika na wodę aby móc lepiej wizualizować zmiany w poziomie wody korzystając z większego zakresu czujnika odległości. Problemem okazało się jednak dobranie odpowiedniego pojemnika, który byłby na tyle wysoki i o takim kształcie aby czujnik ultradźwiękowy był możliwy do zamontowania a zakres jego pomiaru był jak największy. Aktualnie znajdujący się w rzeczywistym układzie pojemnik na wodę jest dosyć płaski w porównaniu do zakładanego, dlatego też zakres pomiarowy czujnika odległości to o 0 do 7 jednostek, przez co grafika wizualizująca pojemnik często ukazuje przeskakującą wartość ilości wody właśnie przez ten mały zakres jednostek. Wada ta nie wpływa jednak na poprawność działania aplikacji tylko na jej przejrzystość.

# Literatura

- [1]
- [2] Jasmin Blanchette, Mark Summerfield. *C++ GUI Programming with Qt 4*. <http://www.qtrac.eu/C++-GUI-Programming-with-Qt-4-1st-ed.zip>, 2006.
- [3] Analog Devices. AD1871 – stereo audio. [http://www.analog.com/en/prod/0,,760\\_789\\_AD1871%2C00.html](http://www.analog.com/en/prod/0,,760_789_AD1871%2C00.html).
- [4] J. Hallam. *Intelligent Automatic Interpretation of Active Marine Sonar*. Praca doktorska, University of Edinburgh, Edinburgh, 1984.
- [5] UM Inc. User modeling and user adapted interaction. <http://www.um.org/>.
- [6] K. Klinger. Informationsverarbeitung durch menschen. *NTZ*, 12(2):73–82, 1998.
- [7] R. Kuc. Forward model for sonar maps produced with the polaroid ranging module. *IEEE Trans. Robot. Automat.*, 19(2):358 – 362, Marzec 2003.
- [8] A. Litwin, J. Kowalski, R. Nowak, J. Ziobro. Transport phenomena in InSb doped with various impurities. *11 International Conference on the Physics of Semiconductors*, wolumen 2, strony 952 – 957, Warszawa – Poland, July 25 – 29 1972.
- [9] M. Michnikowski. Systemy wspomagania decyzji. R. Rudkowski, redaktor, *Informatyka medyczna*. PWN, Warszawa, 2003.
- [10] R. Rudowski, redaktor. *Informatyka medyczna*. PWN, Warszawa, 2003.
- [11] M. W. Spong, M. Vidyasagar. *Dynamika i sterowanie robotów*, rozdział 9, strony 266 – 269. WNT, Warszawa, 1989.
- [12] J. Szabatin. *Podstawy teorii sygnałów*. WKŁ, Warszawa, 2000.
- [13] AAMAS-04 Workshop. Modeling others from observations. <http://www.cs.biu.ac.il/~galk/moo2004/>.
- [14] AAMAS-04 Workshop. Modeling others from observations. <http://www.cs.biu.ac.il/~galk/moo2004/>.