

	PRZEDMIOT:	PRZEDMIOT WYBIERALNY XIV: NARZĘDZIA INFORMATYCZNE W PRAKTYCE INŻYNIERSKIEJ		
	KIERUNEK STUDIÓW:	ELEKTROTECHNIKA	ROK STUDIÓW:	V
	ROK AKADEMICKI:	2021/2022	SEMESTR:	3
	TEMAT:	Wykonanie programu służącego do symulacji funkcjonowania stacji pogodowej mierzącej temperaturę, wilgotność itp. oraz przedstawienia tych danych za pomocą LabView		
IMIE:	KACPER	DATA WYKONANIA ĆWICZENIA:		2022-06-03
NAZWISKO:	HOFFMAN	DATA ODDANIA SPRAWOZDANIA:		2022-06-13
OCENA:	DATA:	UWAGI		

# 1 Projekt 1

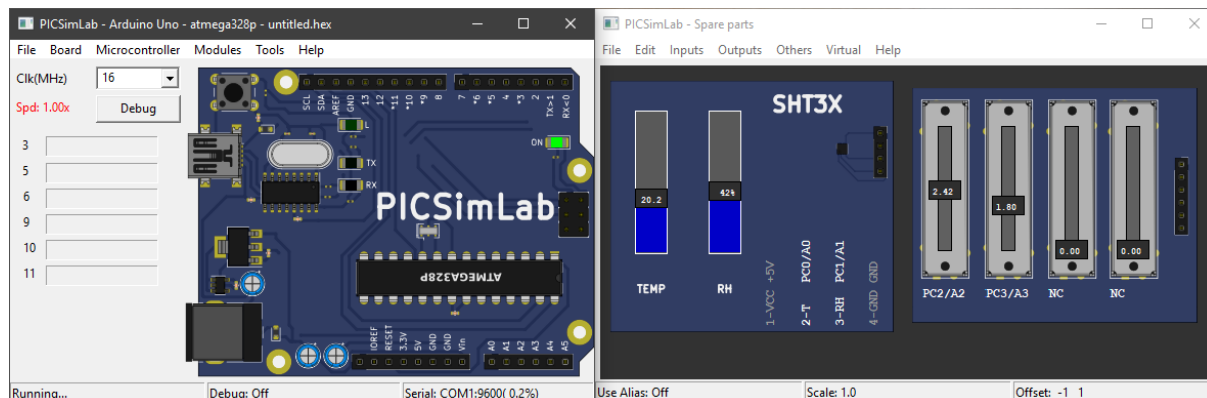
## 1.1 Cel Projektu

Celem projektu było stworzenie w oprogramowaniu LabView, Arduino oraz PicSimLab symulacji funkcjonowania stacji pogodowej. Wielkości pomiarowe symulowane zostały przy pomocy wirtualnej płytki Arduino Uno w PicSimLab, dane owe były odczytywane skryptem napisanym w Arduino IDE a następnie przekazywane do analizy w LabView. Symulowana stacja zabezpieczona jest hasłem, bez którego odczyt danych się nie odbędzie. Po otrzymaniu parametrów program automatycznie przedstawi je liczbowo oraz na wykresach. W ramach pracy zasymulowano odczyt temperatury, wilgotności względnej powietrza, ciśnienia atmosferycznego oraz zapylenia. Wartości aktualne, średnie minutowe oraz średnie godzinne wyświetlone będą na odpowiednich przebiegach oraz zostaną pokazane numerycznie. Wykresy można wyczyścić, a ich przebiegi zapisać.

## 1.2 Algorytm Programu

Algorytm programu składa się z kilku części. W PicSimLab ustawiono połączenie płytki wirtualnej z portem COM1, a dzięki opcji Spare Parts zasymulowano dwa dodatkowe komponenty: SHT3X oraz potencjometry. SHT3X to połączony miernik temperatury i wilgotności powietrza. Pomiar temperatury odbywa się w zakresie od -40 do 125 C, a z kolej pomiar wilgotności względnej jest od 0 do 100 %. Potencjometry z kolej złużą do symulacji parametrów do których mierników nie posiada płytka Arduinio Uno. Są nimi pomiar ciśnienia oraz pomiar zapylenia powietrza. Pomiar ciśnienia założono, że odbywa się od 950 do 1050 hPa. Z kolej pomiar zapylenia wykonuje się od 0 do 200  $\mu\text{g}/\text{m}^3$ . Dodane komponenty podłączono kolejno do portów analogowych PC0/A0, PC1/A1, PC2/A2 oraz PC3/A3. Należy pamiętać o tym, że wartości mierzone są wartościami odpowiednimi dla pracy wewnętrznej czujników. Na przykład potencjometry dają wielkości w skali od 0 do 1023, a nie np. od 0 do 200 które oczekujemy. W związku z tym w programie końcowym należy je przeskalować do skali rzeczywistej. Konfiguracja PicSimLab przedstawiona została dalej [1].

Kolejna część pracy wykonana została w środowisku programistycznym Arduinio. Tutaj następuje odczyt wartości z PicSimLab przy pomocy dostosowanego skryptu. W funkcji setup ustawiamy prędkość przesyłu danych na 9600 bitów na sekundę. W funkcji loop z kolej odczytujemy dane. Wielkości z czujników można przypisać do zmiennych sensorA0, sensorA1, sensorA2 i sensorA3. Następnie odczytane wielkości przekazuje się do portu serialnego, tutaj ustawionego na COM0. To podejście umożliwia potem czytanie danych z Arduino w LabView. Algorytm z Arduinio pokazany został poniżej [2].



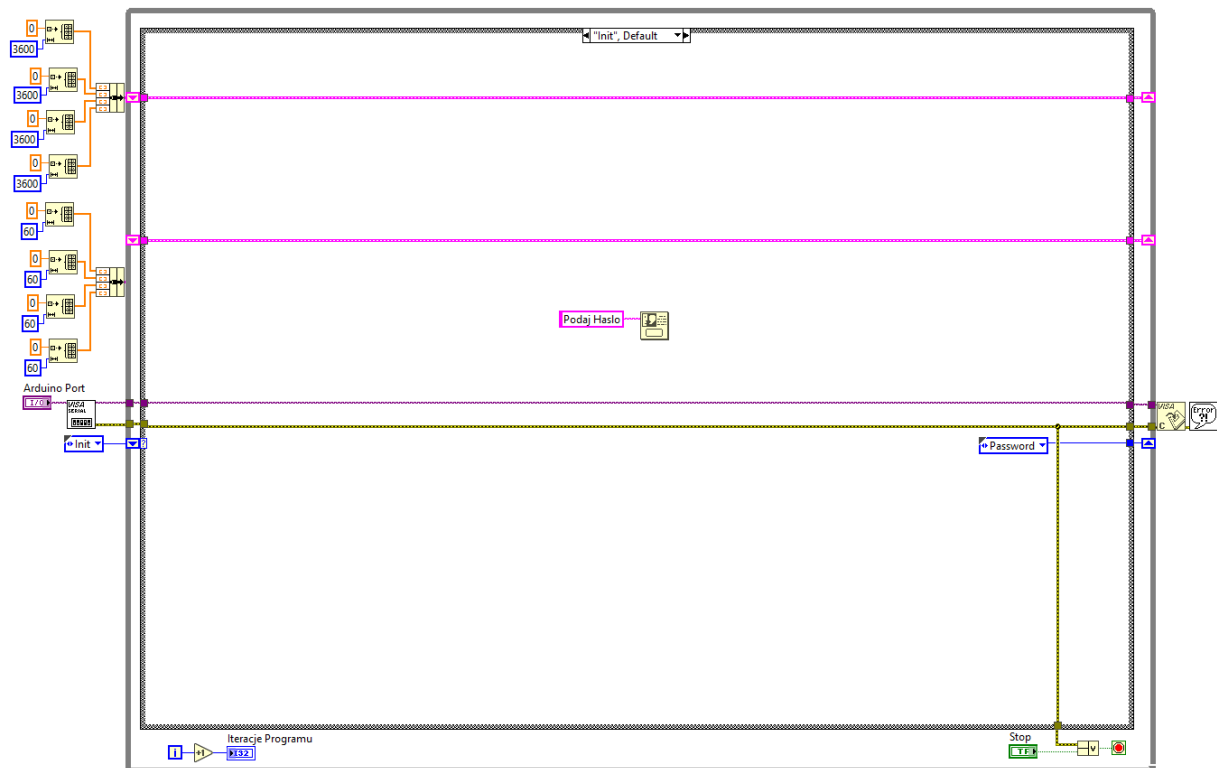
Rysunek 1: Konfiguracja wykonana w PicSimLab

```
void setup()
{
    Serial.begin(9600);
}

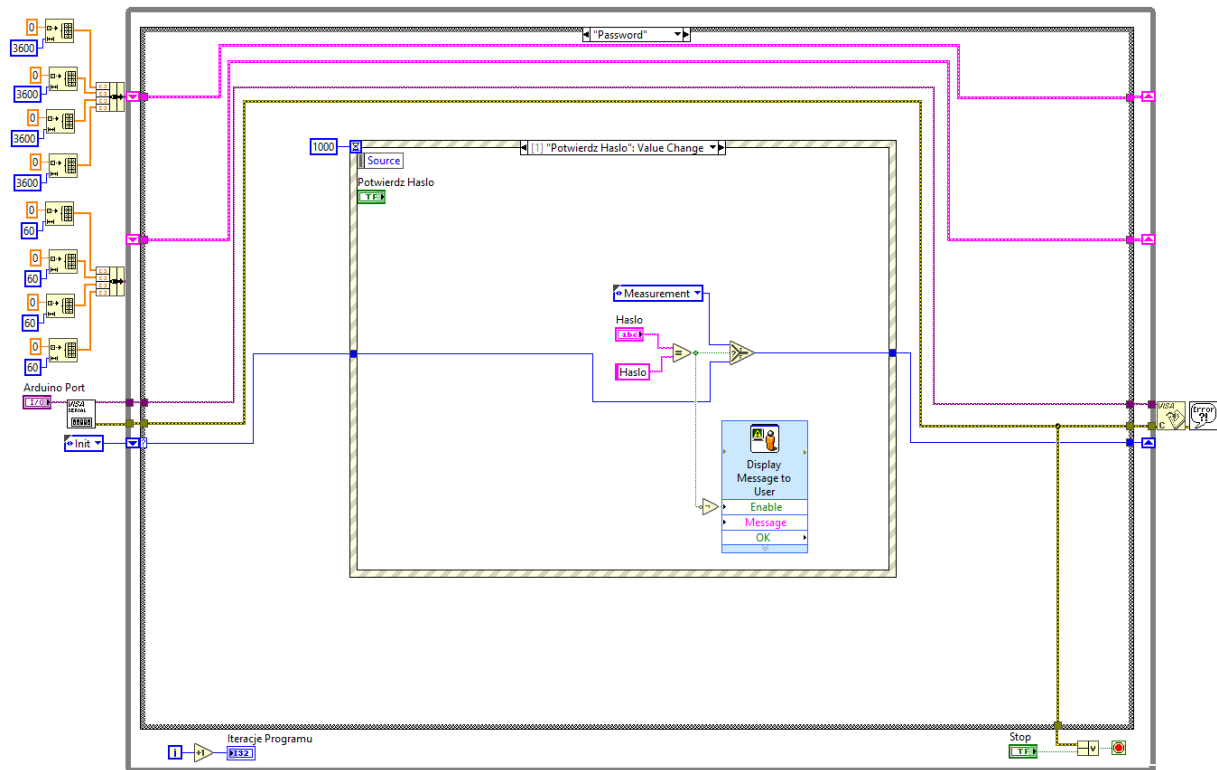
void loop()
{
    int sensorA0 = analogRead(A0);
    int sensorA1 = analogRead(A1);
    int sensorA2 = analogRead(A2);
    int sensorA3 = analogRead(A3);
    Serial.print("T: ");
    Serial.print(sensorA0);
    Serial.print(", RH: ");
    Serial.print(sensorA1);
    Serial.print(", p: ");
    Serial.print(sensorA2);
    Serial.print(", PM10: ");
    Serial.print(sensorA3);
    Serial.print("\n");
    delay(1000);
}
```

Rysunek 2: Algorytm programu w Arduino

Główna część pracy wykonana została w LabView. Tutaj zawarto maszynę stanu do zabezpieczenia hasłem, odczytu danych z Arduino, obliczenia wartości średnich mierzonych parametrów oraz operacji na wykresach. Program na samym początku łączy się z portem Arduino dzięki komponentom VISA. Pierwszym stanem w maszynie stanu jest stan Init. Tutaj (przed pętlą) inicjalizuje się używane później tablice danych oraz informuje się użytkownika o potrzebie wpisania hasła. Tuż po tym program przechodzi do stanu Password. W stanie Password posiadamy strukturę Event. Ona wykrywa, czy został naciśnięty przycisk Potwierdź Hasło. Jeśli zostało, to następuje proste sprawdzenie poprawności. Jeśli tekst wpisany w pole Hasło jest równy aktualnie ustawionemu hasłu, to komponent Select przekazuje dalej stan Measurement. Jeśli nie, użytkownik poinformowany zostaje o nieprawidłowym hasle i pozostaje w stanie Password. Stany Init oraz Password przedstawiono poniżej [3] [4].

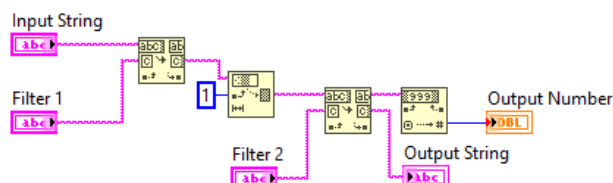


Rysunek 3: Stan inicjalizacyjny programu w LabView

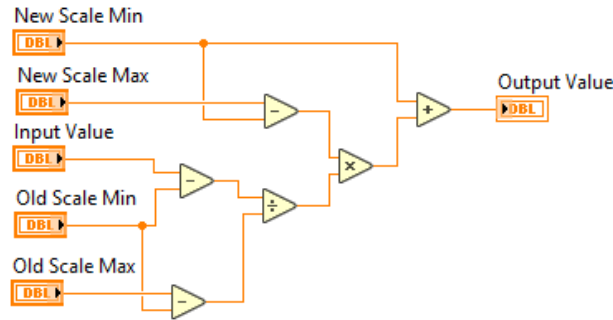


Rysunek 4: Stan hasła programu w LabView

Stan Measurement odpowiada właściwemu odbiorowi i interpretacji danych z Arduino. Tutaj sygnał VISA dociera do elementu Open, który umożliwia odczyt danych jako tekst. W tej części programu skorzystano z czterech podprogramów: ExtractValue, RescaleValue, AverageValue oraz MassAverages. Wewnątrz ExtractValue widoczny jest proces separacji wartości liczbowej z tekstu. W skrypcie Arduino można zauważyć że każda wielkość znajduje się pomiędzy dwoma znakami. Z tego powodu używa się elementu Search/Split String aby usunąć tekst przed liczbą, Substring do ignorowania znaku oddzielającego a na koniec drugi Search/Split String aby oddzielić od tekstu dalszego. Uzyskaną liczbę w formacie tekstu zamienia się na liczbę właściwą. Jako wyjście podprogramu mamy uzyskaną liczbę oraz pozostały tekst. Podprogram RescaleValue służy do naprawienia przedziałów czujników. Zamiana danej liczby ze starej skali na nową odbywa się zgodnie z łatwym do obliczenia wzorem matematycznym (1). AverageValue pozwala na wyznaczenie wartości średniej z zawartości tablicy. Pobiera od tablicę, jej wielkość oraz kolejny element wstawiany do tablicy. Elementy tablicy przesuwają się aby zrobić miejsce na nowy element. Potem dodaje się go w pierwszy indeks i usuwa ostatni. Po tym wstawieniu oblicza się wartość średnią sumując wszystkie elementy i dzieląc przez ilość. MassAverages to typowy podprogram służący do zmniejszenia zużycia miejsca w programie głównym. Tutaj jedyną operacją to jednocześnie wywołanie podprogramu AverageValue na zbiorze czterech tablic z czterema ilościami i czterema nowymi elementami. Zastosowanie operacji pakowania bundle i rozpakowania unbundle umożliwia znaczne uproszczenie algorytmu programu głównego. Wszystkie podprogramy przedstawiono poniżej [5] [6] [7] [8].

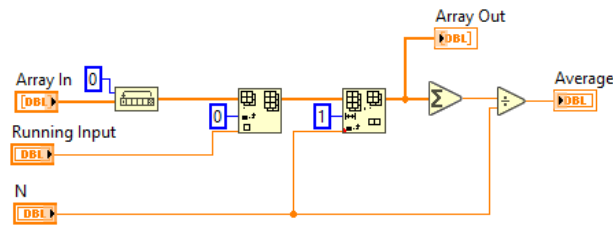


Rysunek 5: Podprogram ExtractValue w LabView

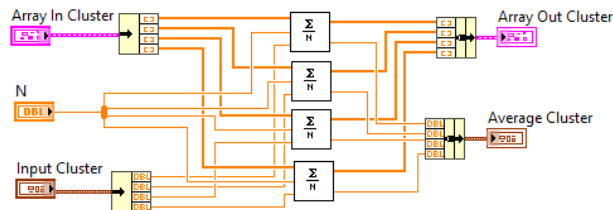


Rysunek 6: Podprogram RescaleValue w LabView

$$x_{nowy} = skala_{minnowy} + (skala_{maxnowy} - skala_{minnowy})(x_{stary} - skala_{minstary}) / (skala_{maxstary} - skala_{minstary}) \quad (1)$$



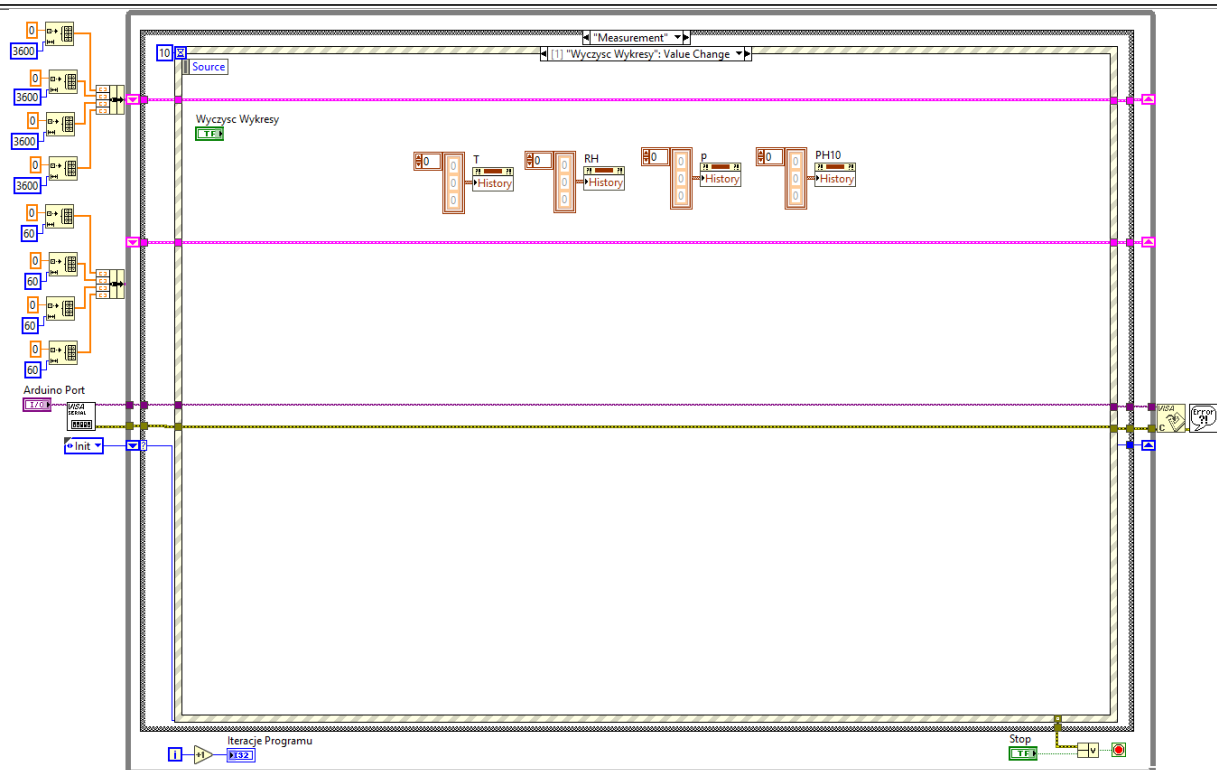
Rysunek 7: Podprogram AverageValue w LabView



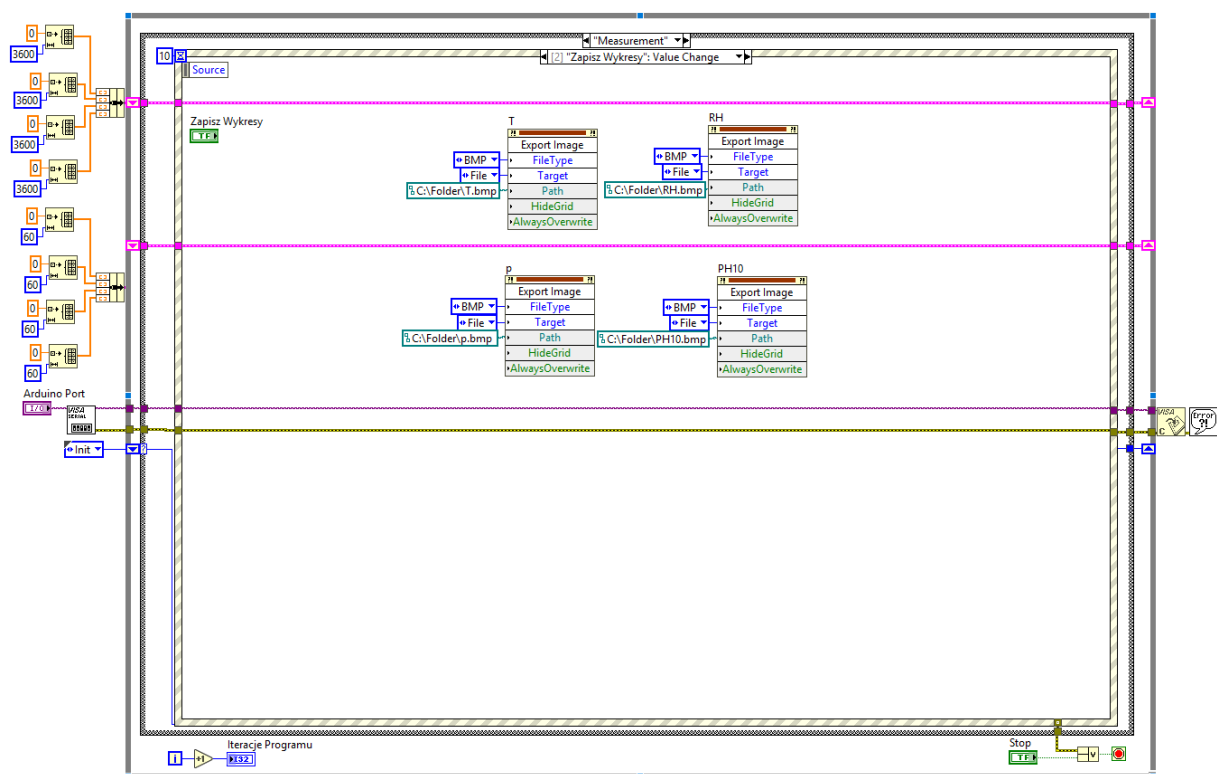
Rysunek 8: Podprogram MassAverages w LabView

Stan Measurement używa powyższych podprogramów w kilku miejscach. Najpierw czterokrotnie wywołano ExtractValue. Tym sposobem pod rząd uzyskano kolejno wartości temperatury, wilgotności, ciśnienia i zapylenia ignorując całkowicie tekst pomiędzy nimi. Te dane trafiają potem do elementów RescaleValue, zamieniając wielkości ze skali czujnika na skalę właściwą dla danej wielkości. Wyznaczone wielkości właściwe przekazywane są do wyświetlenia. W następnym kroku należy obliczyć średnie wielkości. Wcześniej zainicjalizowane tablice wstawia się jako główny argument podprogramu MassAverages. Jako parametr drugi jest ilość elementów. Sprawdzając ilość wykonanych iteracji możemy uzyskać wiarygodne wielkości średnie nawet jeśli nie minęła minuta lub godzina. W tym celu wykonuje się proste sprawdzenie: Czy ilość iteracji jest mniejsza od 60 lub 3600 sekund. Jeśli tak, to używamy ilość iteracji, a w przeciwnym przypadku liczbę stałą. Jako ostatnie wejście wstawiamy parametry aktualne spakowane ze sobą za pomocą operacji bundle. Obliczanie średnich wykonuje się w podprogramach, a na koniec otrzymujemy spakowane wyniki na wyjściu. Rozpakowujemy je za pomocą unbundle i wyświetlamy. Na sam koniec bierzemy obliczone wielkości średnie i przedstawiamy je wspólnie na wykresach. Pełny algorytm przedstawiono poniżej [9].





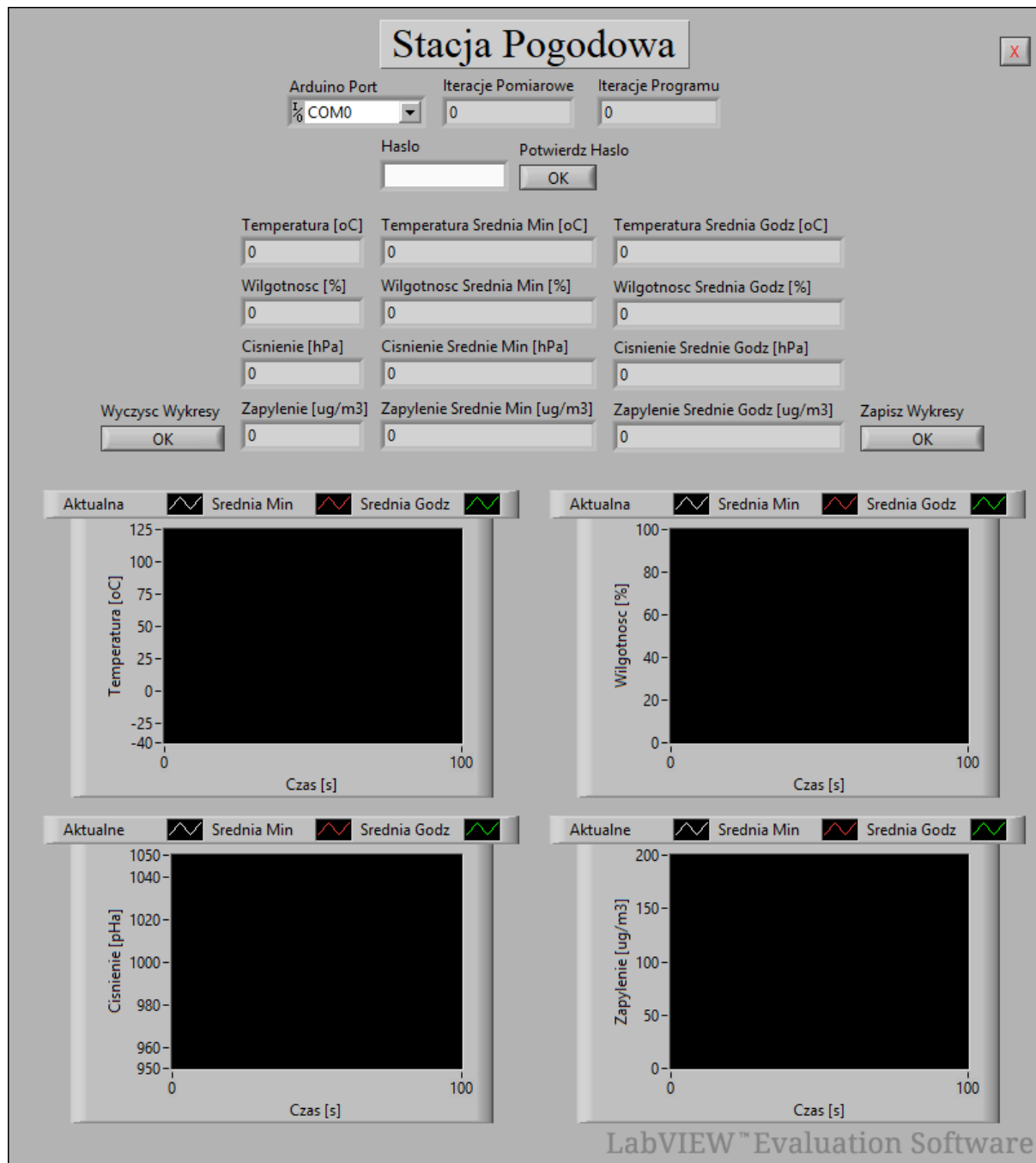
Rysunek 10: Czyszczenie wykresów w LabView



Rysunek 11: Zapisanie wykresów w LabView

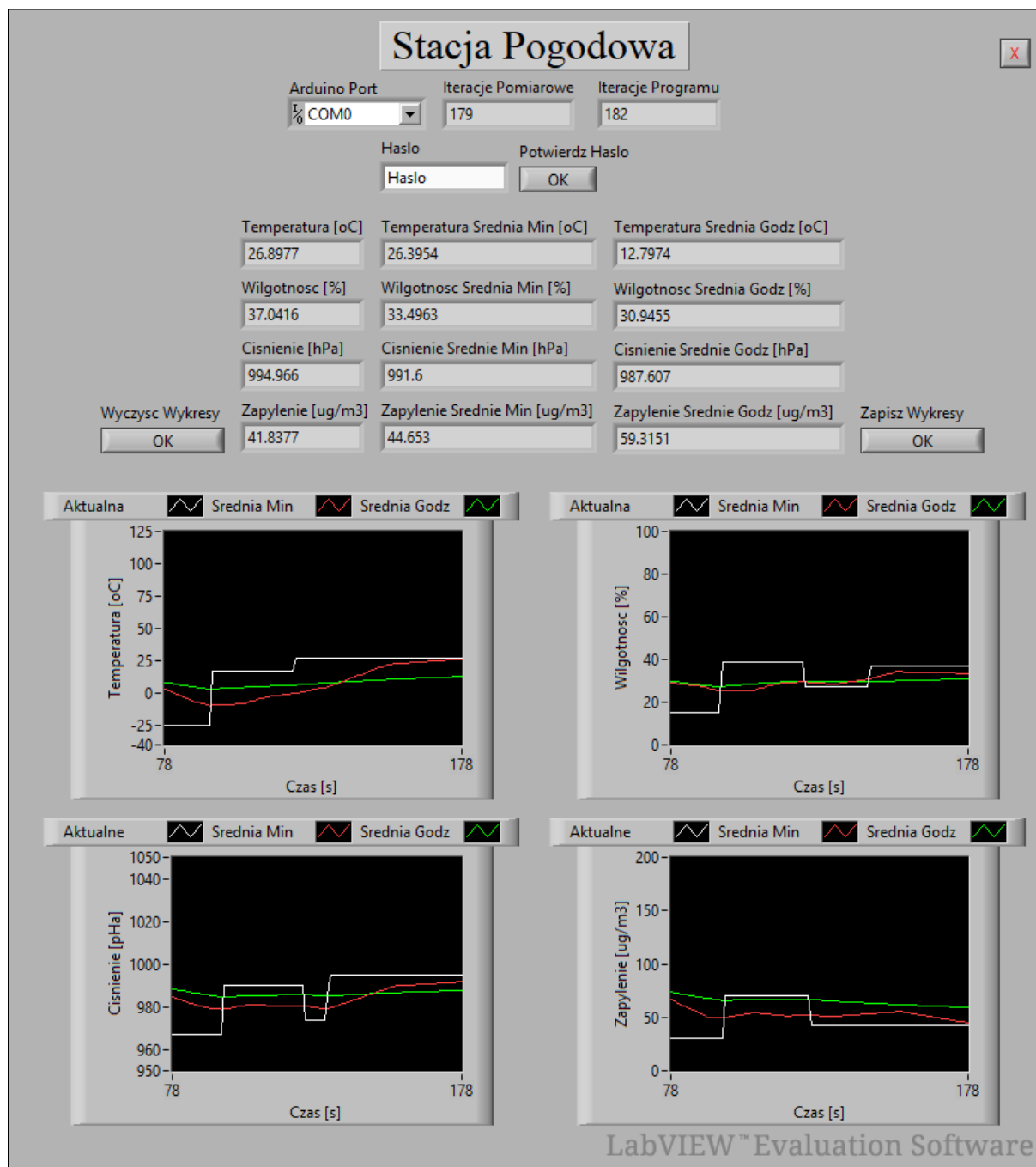
### 1.3 Opis Programu

Jak przedstawiono powyżej, działanie programu polega na symulacji parametrów pogodowych, wykonaniu obliczeń a następnie wyświetleniu wyników na odpowiednich komponentach. Jako parametry główne uznajemy tutaj temperaturę, wilgotność powietrza, ciśnienie atmosferyczne, zapylenie oraz średnie minutowe/godzinowe tych wielkości. One wszystkie zostały wyświetlone i liczbowo i graficznie. Oprócz nich mamy oczywiście pole na hasło wraz z przyciskiem. Na samym początku program prosi o wprowadzenie hasła. Jeśli hasło wpisane jest nieprawidłowe, dostaje się informację o tym. Jeśli jest, to program przechodzi dalej. Wielkości symulowane przez PicSimLab odbiera Arduino, a następnie przechodzą one do LabView, gdzie są one wyświetlane. Wykresy pokazujące te wielkości można wyczyścić albo zapisać jako pliki BMP w określonym folderze. Okno programu wraz z przykładem działania pokazano poniżej [12] [13].



Rysunek 12: Okno wykonanego programu w LabView





Rysunek 13: Okno działającego programu w LabView

## 1.4 Podsumowanie

Po uruchomieniu programu oraz wykonaniu testów możemy stwierdzić, że program funkcjonuje prawidłowo. Po uruchomieniu użytkownik porszony jest o hasło. Podanie nieprawidłowego podaje informację zwrotną, a prawidłowe przechodzi do pomiarów. Utworzone przez niego przebiegi zgadzają się z symulacją. Zasymulowane temperatury, wilgotności, ciśnienia oraz zapylenia odpowiadają mierzonym w PicSimLab. Dodatkowo, program umożliwia wykonanie wykresów przedstawiających zmienność owych parametrów. Widoczne są na wykresach wielkości aktualne, średnie minutowe oraz średnie godzinne. Funkcja czyszczenia wykresów pozwala wykonywać badania od wybranego stanu początkowego, a zapis wykresów pozwala na późniejszą analizę. Po przełączeniu z PicSimLab na prawdziwą płytkę Arduino z komponentami SHT3X, barometrem i czujnikiem jakości powietrza program może funkcjonować jako stacja pogodowa.