

AKADEMIA GÓRNICZO-HUTNICZA  
IM. STANISŁAWA STASZICA W KRAKOWIE

Wydział Inżynierii Metali i Informatyki Przemysłowej



PRACA DYPLOMOWA INŻYNIERSKA  
pt.

„Opracowanie, budowa oraz  
implementacja systemu pobierającego  
oraz analizującego podstawowe  
informacje o warunkach  
atmosferycznych.”

Imię i nazwisko dyplomanta:

**Patryk Zwolak**

Kierunek studiów:

**Edukacja Techniczno-Informatyczna**

Nr albumu:

**268728**

Promotor:

**dr inż. Piotr Kustra**

Recenzent:

**dr inż. Łukasz Sztangret**

Podpis dyplomanta:

Podpis promotora:

Kraków 2018

„Uprzedzony o odpowiedzialności karnej na podstawie art. 115 ust. 1 i 2 ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (t.j. Dz.U. z 2006 r. Nr 90, poz. 631 z późn. zm.): „Kto przywłaszcza sobie autorstwo albo wprowadza w błąd co do autorstwa całości lub części cudzego utworu albo artystycznego wykonania, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 3. Tej samej karze podlega, kto rozpowszechnia bez podania nazwiska lub pseudonimu twórcy cudzy utwór w wersji oryginalnej albo w postaci opracowania, artystyczne wykonanie albo publicznie zniekształca taki utwór, artystyczne wykonanie, fonogram, wideogram lub nadanie.”, a także uprzedzony o odpowiedzialności dyscyplinarnej na podstawie art. 211 ust. 1 ustawy z dnia 27 lipca 2005 r. Prawo o szkolnictwie wyższym (t.j. Dz. U. z 2012 r. poz. 572, z późn. zm.) „Za naruszenie przepisów obowiązujących w uczelni oraz za czyny uchylające godności studenta student ponosi odpowiedzialność dyscyplinarną przed komisją dyscyplinarną albo przed sądem koleżeńskim samorządu studenckiego, zwanym dalej "sądem koleżeńskim"", oświadczam, że niniejszą pracę dyplomową wykonałem(-am) osobiście i samodzielnie i że nie korzystałem(-am) ze źródeł innych niż wymienione w pracy.”

Kraków, dnia .....

Podpis dyplomanta.....

*Pragnę serdecznie podziękować promotorowi  
dr inż. Piotrowi Kustrze za pomoc w realizacji,  
wsparcie, wyrozumiałość oraz poświęcony czas nad tą pracą  
Bardzo dziękuję za wszystkie cenne rady i wskazówki.*

## Spis treści

1. Wstęp .....	5
2. Założenia i cel pracy .....	6
3. Zastosowane podzespoły .....	7
3.1. Raspberry Pi 3.....	7
3.1.1. Charakterystyka urządzenia.....	7
3.1.1. System operacyjny .....	8
3.2. DHT21 (AM2301) – Czujnik temperatury i wilgotności.....	8
3.3. BMP180 – Czujnik ciśnienia .....	9
3.4. BH1750 – Czujnik natężenia światła.....	10
4. Zastosowane technologie, systemy, biblioteki .....	12
4.1. Python – Język programistyczny.....	12
4.2. MySQL – System zarządzania relacyjnymi bazami danych .....	13
4.3. HTML – Język do budowania stron internetowych .....	14
4.4. PHP – Język do budowania stron internetowych oraz aplikacji webowych.....	14
4.5. JavaScript – Język skryptowy rozszerzający HTML o dodatkowe możliwości .....	16
4.6. Biblioteki.....	17
5. Oprogramowanie mikrokontrolera .....	18
5.1. BMP180 .....	18
5.2. DHT21 .....	20
5.3. BH1750 .....	22
6. Oprogramowanie strony internetowej .....	25
6.1. Praca po stronie serwera.....	25
6.2. Tworzenie wykresów .....	29
6.3. Strona internetowa .....	31
7. Testy .....	33
8. Zakończenie .....	36
9. Literatura .....	37

# 1. Wstęp

W teraźniejszych czasach znajomość aktualnych warunków atmosferycznych odgrywa bardzo ważną rolę w życiu codziennym. Do uzyskania takich informacji służą stacje meteorologiczne rozmieszczone w różnych miejscach. Są one urządzeniami wyposażonymi w czujniki temperatury, ciśnienia, wiatromierz, deszczomierz itd. Warto wiedzieć, iż stacje pogodowe znajdują swoje zastosowanie nie tylko w celach domowych. Znajomość aktualnych informacji o temperaturze, wilgotności, ciśnieniu szybkości wiatru oraz jego kierunku znajduje swoje przeznaczenie np. w przemyśle chemicznym, portach morskich i lotniczych, pojazdach wojskowych, rolnictwie czy stokach narciarskich.

Patrząc z perspektywy rozwoju technologii czujniki badające warunki atmosferyczne są coraz częściej stosowane, począwszy od pojazdów samochodowych, stacji meteorologicznych, sterowanie panelami fotowoltaicznymi aż po ochronę techniczną budynku, która opiera się np. na oszczędzaniu energii w przypadku wykrycia wiatru rozwijane są rolety blokujące utratę ciepła.

Pomiary atmosferyczne analizowane w odpowiedni sposób wykorzystywane są do prognoz pogody, które następnie służą do badań procesów fizycznych występujących w klimacie oraz atmosferze. Takie badania mają na celu informowanie przed możliwością pojawienia się niebezpiecznych zjawisk takich dziedzin jak rolnictwo, energetyka, budownictwo czy komunikacja.

## **2. Założenia i cel pracy**

Głównym celem pracy inżynierskiej jest opracowanie, budowa oraz implementacja systemu pobierającego z odpowiednich czujników informacje o warunkach atmosferycznych oraz ich automatyczna analiza i prezentacja przedstawionych wyników w postaci wykresów.

Szczególnym założeniem jest podłączenie do mikrokontrolera (Raspberry Pi 3) czujnika temperatury, wilgotności, ciśnienia oraz natężenia światła oraz stworzenie skryptów pozwalających na realizowanie pomiarów co 50 oraz 3 sekundy na każdy czujnik. Pierwszy rodzaj odczytu jest średnią arytmetyczną sześciu pomiarów wykonanych w odstępach 50 sekundowych, co przekłada się na jeden pomiar w pięciominutowym odstępie czasu. Drugi rodzaj odczytu będzie wyświetlany jako aktualne warunki atmosferyczne panujące w miejscu, gdzie znajduje się urządzenie pomiarowe. W takiej postaci dane wysyłane są do zewnętrznej bazy danych MySQL. Pomiary wykonane w pięciominutowych odstępach, zapisane w bazie danych posłużą do przedstawienia wyników w postaci wykresów. Dzięki zaprojektowaniu strony internetowej przy użyciu technologii PHP, HTML, CSS i JavaScript możliwe będzie wyświetlenie wyników w postaci przyjaznych dla oka wykresów w czasie rzeczywistym jak i z możliwością wyświetlenia historii pomiarów z zakresu dnia, tygodnia oraz miesiąca.

### **3. Zastosowane podzespoły**

Rozdział ten ma na celu opisanie oraz przedstawienie charakterystyki komponentów, z których składa się system pobierający oraz analizujący warunki atmosferyczne. Przedstawione w poniższych podrozdziałach zostaną m.in. czujniki, mikrokontroler Raspberry Pi 3 oraz jego system. Zaprezentowane zostaną rysunki opisujące komponenty oraz ich schemat połączenia, a także tabele z najistotniejszą specyfikacją.

#### **3.1 Raspberry Pi 3**

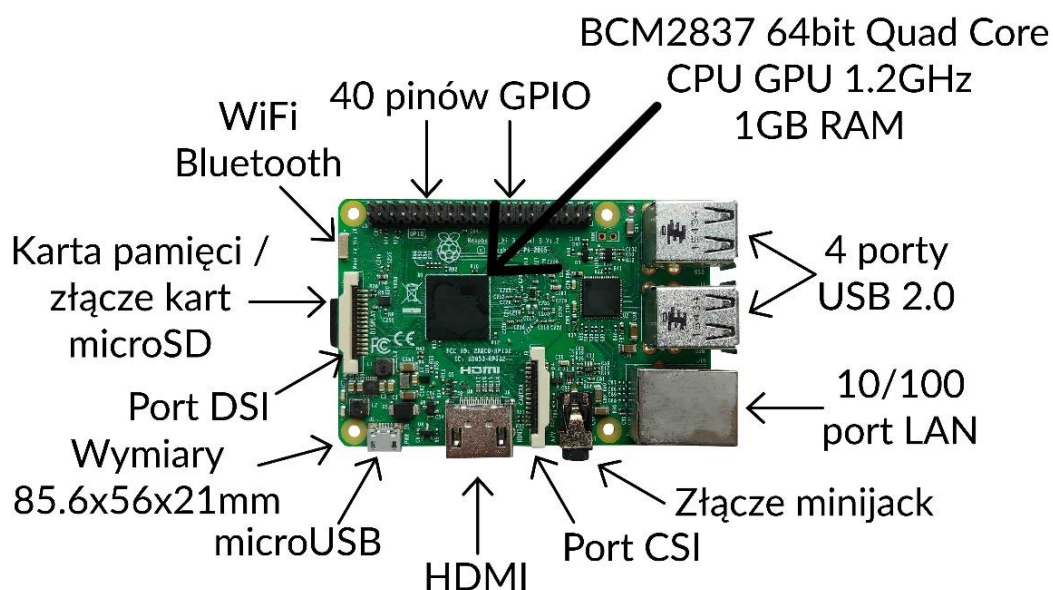
Współczesny rozwój technologii oraz zapotrzebowanie firm i osób indywidualnych stawiających na nowoczesność, przyczynia się do tworzenia coraz mniejszych i szybszych komputerów. Jednym z takich jest jednopłytkowy komputer Raspberry Pi 3. Stworzony został do celów wspierających naukę informatyki. W poniższych podrozdziałach przedstawiona zostanie jego charakterystyka oraz dedykowany system operacyjny.

##### **3.1.1 Charakterystyka urządzenia**

Wielkości karty kredytowej minikomputer Raspberry Pi 3 [1], wyposażony w złącza GPIO jest doskonałym rozwiązaniem do tworzenia niewielkich, przenośnych, mobilnych urządzeń czy robotów. Wyposażony został w czterordzeniowy procesor ARM-8 Cortex-A53 o taktowaniu 1.2GHz. Pamięć RAM o wielkości 1GB, umożliwia działanie dużej ilości procesów w tle. Dodatkowo został zaopatrzony w cztery porty USB, jeden port Ethernet, bezprzewodową kartę sieciową WiFi oraz Bluetooth. Gniazdo HDMI w wersji 1.4 pozwala na podłączenie monitora w celu zarządzania systemem z poziomu interfejsu graficznego lub za pomocą terminala poleceń. Do zasilenia urządzenie potrzebuje napięcia 5,1 V oraz natężenia od 1 – 2,5 A. Wymiary minikomputera wynoszą 55,6 x 85 x 21 mm. Urządzenie mimo swojego jednopłytkowego formatu nie zostało wyposażone w dysk wewnętrzny, natomiast posiada możliwość podłączenia karty microSD. Takie rozwiązanie pozwala na dostosowanie pamięci w zależności od potrzeb użytkownika.

GPIO – (General Purpose Input/Output) – złącza wejścia / wyjścia pozwalające na sterowanie urządzeniami peryferyjnymi, czujnikami oraz innymi podzespołami

kompatybilnymi z mikrokontrolerem. Dzięki GPIO możliwa jest komunikacja między czujnikami a Raspberry. Charakterystykę mikrokontrolera podano na rysunku 1.



Rysunek 1. Opis podzespołów mikrokontrolera Raspberry Pi 3.

### 3.1.2 System operacyjny

Podstawowym systemem operacyjnym mikrokontrolerów Raspberry Pi jest Raspbian [2]. System ten oparty jest na Debianie, który z kolei jest dystrybucją Linuxa. Raspbian jest systemem dostosowanym pod podzespoły mikrokontrolera czyniąc go łatwym w obsłudze oraz wszechstronnym urządzeniem. Ponadto mikrokomputer zyskał już sporą sławę, co przyczyniło się do powstania większej ilości systemów operacyjnych, m.in.: Windows 10 IoT Core, Ubuntu Mate, OSMC czy PINET. Z powodzeniem można stwierdzić, iż odpowiedni system operacyjny czyni z tego urządzenia mały komputer z dużą ilością narzędzi.

### 3.2 DHT21 (AM2301) – Czujnik temperatury i wilgotności

Cyfrowy czujnik DHT21 [3] nazywany również AM2301 został wyposażony w pojemnościowy czujnik wilgotności i termistor. Posiada napięcie zasilania od 3,3 V do 5,5. Pobór prądu urządzenia wynosi średnio 1,5mA na każdy pomiar, natomiast jego wymiary to 22 x 28 x 5 mm. Warto wspomnieć, iż małe rozmiary, niskie zużycie energii, wysoka dokładność, zasięg transmisji sięgający do 20 metrów oraz niski koszt czyni

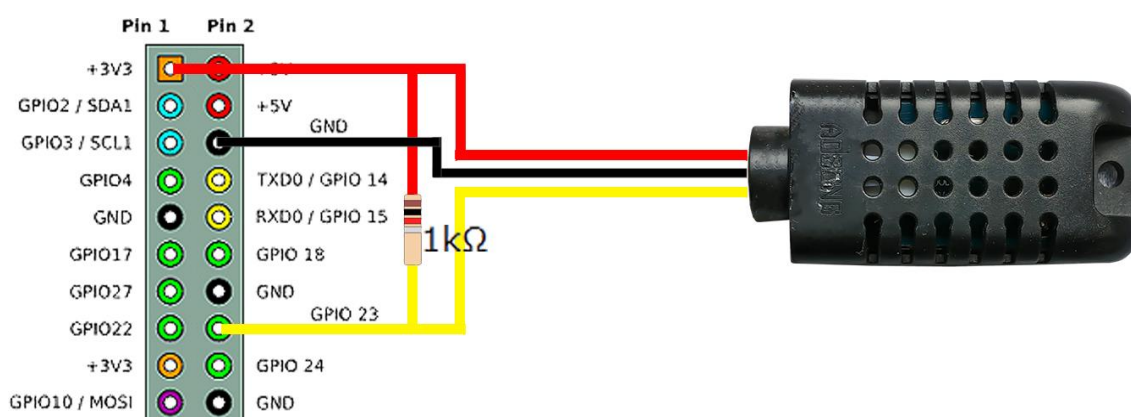


czujnik doskonałym rozwiązaniem do zastosowań w urządzeniach do pomiaru warunków atmosferycznych. W tabeli 1 przedstawiona została charakterystyka czujnika DHT21.

Tabela 1. Charakterystyka czujników wilgotności i temperatury.

	Czujnik wilgotności	Czujnik temperatury
Zakres pomiaru:	0 - 100% RH	-40°C - +80°C
Dokładność:	+/- 3% RH	+/- 0,5°C
Czas odpowiedzi:	około 2s	około 2s
Rozdzielczość:	0,1% RH	0,1°C

Na rysunku 2 przedstawiony został schemat połączenia czujnika z GPIO.



Rysunek 2. Schemat połączenia czujnika z GPIO.

### 3.3 BMP180 – Czujnik ciśnienia

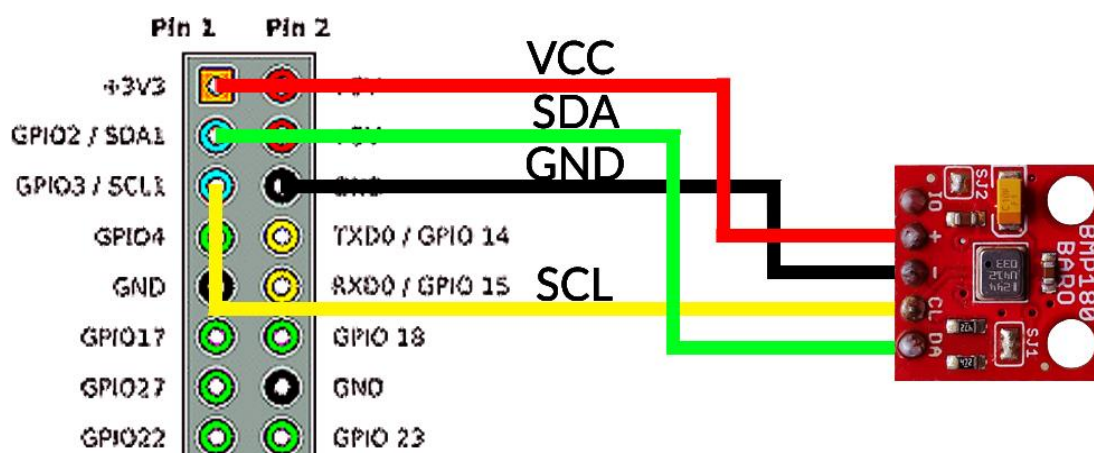
Cyfrowy czujnik ciśnienia (barometr) BMP180 [4] posiada napięcie zasilania z zakresu od 1,8 do 3,6V co sprawia, iż jest idealnym urządzeniem do pracy z mikrokontrolerem Raspberry. Komunikacja odbywa się za pomocą magistrali I2C. Wysoka dokładność, odporność i trwałość czujnika możliwa jest dzięki zastosowaniu technologii piezo-rezystancyjnej. Pobór prądu BMP180 wynosi średnio około 5  $\mu$ A (0,005 mA) na każdy pomiar, natomiast jego fizyczne wymiary wynoszą 16 x 16 mm. Dużą zaletą urządzenia jest jego zaopatrzenie w czujnik ciśnienia, temperatury oraz wysokości. Niestety do prawidłowego pomiaru wysokości niezbędna jest informacja o

aktualnym ciśnieniu panującym na poziomie morza. W skrócie można powiedzieć, iż wysokość jest liczona na podstawie różnic ciśnień. BMP180 wykorzystywany jest do zwiększenia dokładności działania systemów GPS (wykrywanie nachylenia), urządzeń sportowych czy stacji prognozy pogody. W tabeli 2 przedstawiona została charakterystyka tego elementu elektronicznego.

Tabela 2. Charakterystyka czujnika BMP180.

	Czujnik ciśnienia	Czujnik temperatury
Zakres pomiaru:	300 hPa - 1100 hPa	40°C - +85°C
Dokładność:	+/- 0,12 hPa	+/- 1.0°C
Czas odpowiedzi:	4.5 ms – 25.5 ms	

Schemat połączenia BMP180 z GPIO przedstawiony na rysunku 3.



Rysunek 3. Schemat połączenia czujnika ciśnienia z GPIO.

### 3.4 BH1750 – Czujnik natężenia światła

BH1750 [5] jest czujnikiem natężenia światła, którego zakres pomiaru wynosi od 1 do 65535 lx<sup>1</sup>. Cyfrowy czujnik zasilany jest napięciem z zakresu od 3 do 5 V. Komunikacja tak jak w przypadku BMP180 odbywa się przy użyciu magistrali I2C. Dużą zaletą BH1750 jest bardzo mały wpływ światła podczerwonego na wynik pomiaru. Pobór prądu czujnika wynosi średnio około 120 µA (0,120 mA) na każdy

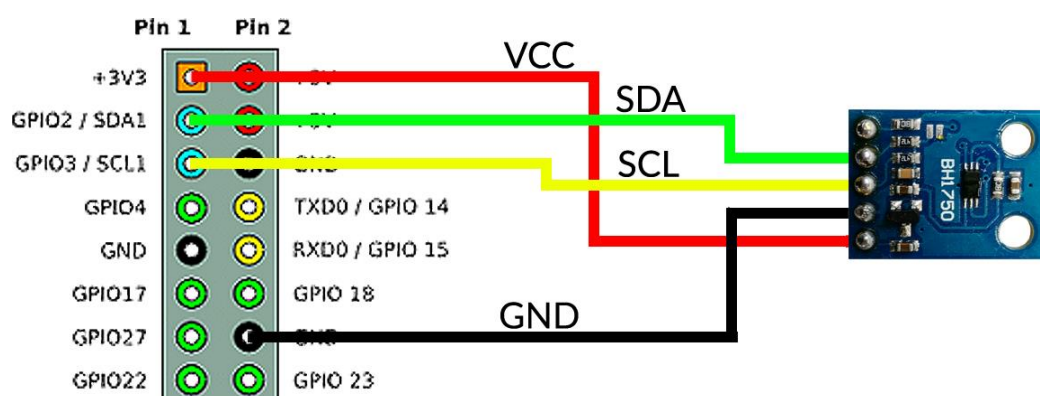
<sup>1</sup> lx (luks) – jednostka natężenia światła [<https://www.translatorscafe.com/unit-converter/PL/illumination/1-4/lux-foot-candle/>]. Luks opisuje się wzorem  $1\text{lx} = 1\text{lm} / 1\text{m}^2$ . Luks można określić jako 1 lumen padający na powierzchnię  $1\text{m}^2$

pomiar, natomiast jego fizyczne wymiary wynoszą 16 x 21 x 3,3 mm. Urządzenie zostało wyposażone w funkcję, której zadaniem jest filtrowanie szumu przemysłowego o częstotliwości 50/60Hz w celu zwiększenia dokładności pomiaru. Ponadto rezultat pomiaru wyników z czujnika jest podobny do charakterystyki widmowej ludzkiego oka. Czujnik swoje zastosowanie znajduje między innymi w urządzeniach mobilnych, wyświetlaczach i telewizorach LCD, kamerach oraz aparatach. Charakterystyka BH1750 przedstawiona została w tabeli 3.

Tabela 3. Charakterystyka czujnika BH1750.

	Czujnik światła
Zakres pomiaru:	1 - 65535 lx
Dokładność:	+/- 1 lx
Czas odpowiedzi:	120 ms

Schemat połączenia BH1750 z GPIO przedstawiony na rysunku 4.



Rysunek 4. Schemat połączenia czujnika natężenia światła z GPIO.

## **4. Zastosowane technologie, systemy, biblioteki**

Celem tego rozdziału jest przedstawienie i opisanie wykorzystanych w niniejszej pracy inżynierskiej technologii, bibliotek, skryptu oraz systemu, przedstawionych w poniższych podrozdziałach. Do wykonania założonych celów niezbędne było skorzystanie z technologii Python, która umożliwiła stworzenie programów pobierających informacje z wszystkich czujników. System zarządzania bazą danych MySQL, konieczny był do zapisania odczytanych pomiarów. Za prezentowanie wyników odpowiedzialna jest strona internetowa, w której niezbędne było wykorzystanie technologii HTML, PHP, JavaScript oraz skryptu chart.js.

### **4.1 Python – Język programistyczny**

Na wstępie należy zaznaczyć, iż Python [6] jest darmowym, powszechnym językiem programowania wykorzystywanym między innymi przez takie „giganty” jak Google oraz YouTube. Jego prostota użycia oraz możliwości sprawiają, że programiści z różnych dziedzin uważają, iż jest doskonałym językiem do tworzenia dużych jak i małych projektów.

Wysoka przejrzystość, jakość oraz integralność sprawia, iż Python jest prosty do odczytania a także zrozumienia, nawet dla osób które nie są twórcą danego programu. Ponadto język ten umożliwia projektowanie programów OOP (Object oriented programming) czyli programowanie zorientowane obiektowo. Dużą uwagę należy zwrócić na wydajność języka, która dzięki swojej prostocie sprawia, że w pełni działający kod często stanowi od około 1/3 do 1/5 mniejszej ilości znaków niż w Javie lub C++. Przekłada się to na mniejszą ilość znaków, które musi odczytać komputer lub osoba wprowadzająca zmiany w kodzie. Natychmiastowe działanie, bez potrzeby kompilacji utwierdzają powyższe słowa odnośnie wydajności.

Wszystkie wyżej wymienione cechy przyczyniły się prawdopodobnie do tego, że domyślnym (lecz nie jedynym) językiem programowania na Raspberry Pi jest właśnie Python.

## 4.2 MySQL – System zarządzania relacyjnymi bazami danych

MySQL [7] jest jedną z najbardziej powszechnych baz danych, która stanowi niezbędną część wielu stron www. Jej głównym zadaniem jest magazynowanie różnych danych, bez których strony internetowe nie miałyby prawa właściwie funkcjonować. Takimi danymi są dla przykładu loginy, hasła, adresy, numery telefonów, daty, konta użytkowników. W skrócie można powiedzieć, iż MySQL jest systemem zarządzania bazą danych (DBMS ang. Database Management System). SQL (ang. Structured Query Language) jest językiem pozwalającym na pisanie zapytań do baz.

Baza danych MySQL nie bez powodu cieszy się ogromną popularnością, jedną z cech która się do tego przyczyniła jest jej elastyczność, możliwość pracy na około 14 platformach takich jak Linux, Windows, Mac OS X, HP-UX itd. sprawia, że programista nie musi martwić się o dopasowanie oprogramowania i z powodzeniem może przejść do działania. Ponadto kompatybilność bazy z takimi językami jak C, C++, Python, Java, PHP i wiele innych, przyczynia się do jeszcze prostszej obsługi i zarządzania bazami. Koniecznie trzeba zaznaczyć, że najważniejszym celem jakim kierowali się twórcy była szybkość działania oraz wydajność, którą MySQL posiada i z tego jest znany oraz ceniony.

Kolejnym atutem jest duża ilość typów danych takich jak wartości łańcuchowe, typy daty i czasu, liczby, zestawy. Powyższe typy danych możemy jeszcze rozbić na kilka różnych typów np.: na bardzo małe liczby całkowite (TINYINT) oraz liczby zmiennoprzecinkowe (FLOAT). Drugim przykładem może być podanie samego roku (YEAR) lub pełnowymiarowej daty (TIMESTAMP). Takie zastosowanie pozwala na użycie tylko tych typów danych, które są potrzebne w danej tabeli, przekłada się to na zmniejszenie rozmiaru i zwiększenie wydajności bazy przy ogromnej ilości danych. Dodatkowo warto wspomnieć o komendzie SHOW, która umożliwia filtrowanie i przeglądanie zawartości tabel, jej wierszy oraz kolumn po odpowiednim sformułowaniu zapytania.

Prostota pisanych zapytań, tworzenia tabel, łatwość komunikacji, integralność z Pythonem czy PHP, ilość dostępnej wiedzy i kursów wpłynęły na podjęcie decyzji, by to właśnie MySQL był systemem zarządzania baz danych, w których zapisywane są wyniki pomiarów.

### **4.3 HTML – Język do budowania stron internetowych**

HyperText Markup Language czyli HTML [8] to w Polskim tłumaczeniu hipertekstowy język znaczników, który wykorzystuje się do tworzenia stron www. Język HTML oparty jest na tak zwanych znacznikach opisujących treść strony internetowej. W dosłownym tłumaczeniu można powiedzieć, iż strona internetowa to plik tekstowy z właściwie napisanym kodem źródłowym zapisanym w pliku z rozszerzeniem .html, który następnie zostaje zinterpretowany i wyświetlony przez przeglądarkę. Przypadkiem nie jest, że HTML to po prostu plik tekstowy. Zanim powstał WorldWideWeb twórcy głowili się jak zrobić, by odpowiednią treść czy obraz mogły zobaczyć osoby z różnymi systemami operacyjnymi. Ze względu na różnice graficznego interfejsu użytkownika (GUI – Graphical User Interface) w różnych systemach operacyjnych dana strona internetowa musiałaby być napisana pod każdy z nich, co stałoby się bardzo problematyczne i czasochłonne. Odpowiedzią był właśnie plik tekstowy, ponieważ każdy system bez problemu potrafi go odczytać, natomiast do jego prawidłowego działania stworzony został program interpretujący dane znaczniki zwany współcześnie przeglądarką internetową.

Każda strona internetowa napisana w HTML musi zawierać trzy podstawowe sekcje <html>, <head>, <body>. Sekcja <html></html> oznacza początek zawartości html dający przeglądarce informacje, że to właśnie tutaj zaczyna się strona. W sekcji <head></head> znajdują się niezbędne informacje m.in. standard kodowania znaków, tytuł, czcionki, opis, język strony, skrypty, css, itd. – sekcja niewidoczna dla osoby odwiedzającej stronę. Natomiast w sekcji <body></body> znajdują się wszystkie widoczne dla osoby przeglądającej treści takie jak hiperłącza, obrazy, teksty, przyciski, pola edycyjne itd.

### **4.4 PHP – Język do budowania stron internetowych oraz aplikacji webowych**

Skryptowy interpretowany język programowania PHP [7] stworzony został do budowania stron www oraz aplikacji webowych. Język ten umożliwia napisanie skryptu w oddzielnym pliku lub między kodem HTML, który wykonywany jest po stronie serwera. Co więcej zawartość kodu PHP czyli zaczynająca się od „<?php i kończąca na ?>” ukryta jest dla użytkownika przeglądającego stronę, co przekłada się na brak

możliwości skopiowania kodu od autora i wykorzystania jego metod na potrzeby własne.

Przed wszystkim należy zwrócić uwagę na praktyczność tego języka. Pierwszą taką cechą może być m.in. brak konieczności implementowania biblioteki. Za drugi przykład posłuży nam to, że zupełnie użyteczny program może zawierać tylko jedną linię kodu. Doskonałym tego dowodem jest wiersz „<?php echo date("Y-m-d H:i:s"); ?>”, który wyświetli na ekranie aktualną datę oraz godzinę. Ponadto kolejnym potwierdzeniem praktyczności tego języka może być opcja zagnieżdżania wywołania funkcji. Na przykład wiersz „\$ciag=substr(md5(microtime()), 0, 7);” zapisze do zmiennej „\$ciag” wyraz składający się z 7 losowych znaków. Dodatkowo PHP posiada tzw. „typowanie słabe” oznacza to, iż nie jest konieczne jawne tworzenie, usuwanie czy rzutowanie zmiennej. Co więcej, w trakcie kończenia pracy skryptu PHP automatycznie zwalnia różne zasoby oraz usuwa zmienne.

Kolejnymi sporymi zaletami tego języka jest dostęp do 200 domyślnych bibliotek, posiadających razem ponad tysiąc funkcji. Dynamiczne tworzenie stron, mechanizm eksploatacji baz danych, pobieranie danych z formularzy HTML, tworzenie i obsługa PDF, ocena możliwości łamania haseł czy mechanizmy przetwarzania kart kredytowych to jedne z nielicznych możliwości jakie oferuje nam PHP.

Dużym potencjałem jest możliwość korzystania nie z jednego a kilku rozwiązań. Takimi rozwiązaniami jest wybór spośród około 30 baz danych takich jak: Unix, Sybase, Oracle, MySQL, Microsoft SQL Server, IBM DB2, Express, dBase, FontBase oraz wiele innych. Programowanie obiektowe oraz proceduralne także możliwe jest w PHP.

Ostatnia ważna zaleta jaka nasuwa się by potwierdzić dlaczego PHP jest tak powszechny i pozytywnie rozpatrywany wśród środowiska programistycznego to jego cena. Język ten jest darmowy, użytkownik nie musi martwić się o koszty nawet w celach komercyjnych. Można pomyśleć, iż taki system sprawia, że język ten jest niedopracowany, natomiast realia wyglądają wręcz odwrotnie. Jego ogólnodostępność sprawia iż luki w zabezpieczeniu czy potencjalne problemy są w szybkim tempie poprawiane.

## **4.5 JavaScript – Język skryptowy rozszerzający HTML o dodatkowe możliwości**

JS [9] – to język skryptowy umożliwiający budowanie programów, które mają na celu zwiększenie potencjału języka HTML o interaktywność, animacje czy różne efekty wizualne. Na wstępie należy zaznaczyć, iż JavaScript oraz Java to dwa różne języki. Dla przykładu JS jest językiem interpretowanym na PC klienta, natomiast Java wykonywany jest przy użyciu wirtualnej maszyny na PC klienta. Kod JS umieszczany jest w kodzie HTML, natomiast kod Java jest niezależny od HTML i zapisywany jest jako oddzielny plik. W JS nie ma konieczności deklarowania zmiennych, odwrotnie jest w przypadku Javy gdzie zmienne deklarowane są już przed ich użyciem.

JavaScript pozwala m.in. na zamieszczenie w kodzie HTML instrukcji zegara, który bez konieczności odświeżania całej strony będzie odliczał czas, stworzenie bloku automatycznej animacji na stronie, interaktywnych przycisków czy gry przeglądarkowej. Analizując powyższe przykłady można stwierdzić, iż JavaScript jest niepotrzebny ponieważ PHP też umożliwia tworzenie takich animacji czy aplikacji, natomiast byłoby bezsensowne np. wysyłanie informacji na serwer z prośbą o odświeżenie wyświetlanej godziny. Ponadto po takiej odpowiedzi cała strona musiałaby być na nowo odświeżona. Dzięki JS wszystkie operacje takiego typu obliczane są po stronie osoby przeglądającej witrynę (przy użyciu PC oraz przeglądarki) bez konieczności odświeżania całej strony lecz tylko jej elementu przyczyniając się do lepszej przejrzystości strony internetowej. Ponadto takie zastosowanie wpływa na bardzo dużą szybkość działania oraz błyskawiczną reakcję. Doskonałym przykładem witryny wykorzystującej JS jest Google Maps (<https://www.google.pl/maps>). Użytkownik ma do dyspozycji kilka funkcji takich jak przybliżanie i oddalanie mapy, przesuwanie, zmianę widoku itd.



## 4.6 Biblioteki

Do realizacji niniejszej pracy wykorzystano biblioteki umożliwiające m.in. obsługę baz danych, obsługę czujników, daty oraz czasu, obsługę funkcji matematycznych oraz inne, które szczegółowo zostały przedstawione i wytłumaczone poniżej:

- **MySQLdb** - jedna z najbardziej znanych oraz najlepszych bibliotek do obsługi bazy danych MySQL w języku programistycznym Python,
- **Adafruit\_BMP.BMP085** – biblioteka do obsługi czujnika ciśnienia,
- **time** oraz **datetime** – biblioteki do obsługi daty i czasu,
- **subprocess** – biblioteka pozwalająca m.in. na uruchomienie zewnętrznego programu oraz pozyskiwaniu z niego danych wyjściowych,
- **linecache** – biblioteka umożliwiająca m.in. czytanie oraz zapisywanie treści w pliku tekstowym,
- **sys** – biblioteka zapewniająca dostęp do różnych zmiennych oraz funkcji obsługiwanych przez interpreter,
- **smbus** – pozwala na dostęp do interfejsu I2C,
- **jQuery** – biblioteka rozszerzająca możliwości języka JavaScript oraz ułatwiająca korzystanie z niego,
- **chart.js** – **skrypt** w języku JavaScript stworzony dla programistów oraz projektantów w celu możliwości przedstawiania wyników w postaci elastycznych, czytelnych wykresów. Chart.js jest w pełni darmowe, umożliwia wykorzystanie 8 typów wykresów jest w pełni responsywne oraz współpracuje z każdą przeglądarką. Skrypt ten umożliwia tworzenie wykresów: liniowych, słupkowych, radarowych, ciasteczkowych, bąbelkowych itp. oraz mieszaniu ich ze sobą. W swojej pracy inżynierskiej posłużyłem się wykresami słupkowymi oraz liniowymi. Ponadto funkcjonalność skryptu w prosty sposób pozwala na dostosowanie przez programistę takich parametrów jak czcionka, kolory, animacje czy możliwość responsywności,

## 5. Oprogramowanie mikrokontrolera

Podłączenie czujników do GPIO mikrokontrolera Raspberry Pi 3 według schematów zamieszczonych w rozdziale trzecim jest pierwszą czynnością umożliwiającą prawidłowy pomiar warunków atmosferycznych. Kolejną czynnością jest stworzenie programów, które umożliwią odczyt pomiarów. W poniższych podrozdziałach szczegółowo zostały opisane i przedstawione elementy kodu takich programów.

### 5.1 BMP180

Do prawidłowego działania oraz pobierania informacji z czujnika niezbędne jest właściwe oprogramowanie. Wszystkie poniżej przedstawione listingi kodu są zawarte w jednym pliku. Rozdzielone zostały dla łatwiejszej interpretacji. Import bibliotek oraz połączenie bazy danych przedstawione w listingu 1.

```
import MySQLdb
import Adafruit_BMP.BMP085 as BMP085
import time
sensor = BMP085.BMP085()
db=MySQLdb.connect("mysql.agh.edu.pl","pzwolak","xxxxxxxxxxxxxxxxxx","pzwolak")
```

Listing kodu 1. Importowanie odpowiednich bibliotek oraz stworzenie połączenia z bazą danych.

Powyższy listing przedstawia import niezbędnych do działania bibliotek. Ostatnia linia kodu przekazuje do zmiennej „db” informację o połączeniu bez którego nie byłoby możliwe wysyłanie informacji poza urządzenie. Funkcja „MySQLdb.connect()” przekazuje informacje takie jak („nazwa hosta”, „nazwa użytkownika”, „hasło”, „nazwa bazy danych”).

Kod przedstawiony w listingu 2 służy do pobierania z czujnika wartości temperatury, ciśnienia oraz wysokości. Pętla wykonuje się 6 razy co 50 sekund. Funkcja „format(sensor.read\_pressure()\*0.01)” służy do wywołania oraz pobrania informacji o ciśnieniu. Po wykonaniu tej linii kodu funkcja zapisuje do zmiennej „x” wartość ciśnienia w paskalach, dlatego by otrzymać wynik w hPa został on pomnożony o 0.01. Wiersz „x=float(x)” zamienia daną wartość na zmiennoprzecinkową, następnie wynik zostaje zaokrąglony do jednego miejsca po przecinku poleceniem „cis=round(x,1)”. Otrzymany wynik nie jest dokładny, z niewiadomych przyczyn wynik ciśnienia zaniżany jest o 28 hPa dlatego kalibrowany został poleceniem „cis=float(cis+28)”.

Wynik zapisywany jest w tablicy poleceniem „*tab.append(cis)*”. Pozostałe pomiary wysokości oraz temperatury wykonane są w sposób analogiczny do powyższego z inną nazwą zmiennych i funkcji: „*format(sensor.read\_altitude())*” – dla wysokości oraz „*format(sensor.read\_temperature())*” – dla temperatury. Następnie liczona jest średnia arytmetyczna pomiaru z danego czujnika. Wyniki w takiej postaci wysyłane są do bazy danych przy użyciu poniższego kodu przedstawionego w listingu kodu 3.

```
while licz<6:
    x=format(sensor.read_pressure()*0.01)
    x=float(x)
    cis=round(x,1)
    cis=float(cis+28)
    tab.append(cis)

    y=format(sensor.read_altitude())
    y=float(y)
    alt=round(y,1)
    alt=float(alt+71)
    tabb.append(alt)

    z=format(sensor.read_temperature())
    z=float(z)
    temp1=round(z,1)
    temp1=float(temp1)
    tabbb.append(temp1)

    licz+=1
time.sleep(50)
```

Listing kodu 2. Pętla wywołująca funkcje pobrania danych z czujnika BMP180 oraz ich zapisania w tablicy.

```
print('Cisnienie: ', cis)
print('Wysokosc: ', alt)
print("Temperatura: ", temp1)
xt=time.strftime('%Y-%m-%d %H:%M:%S')
cursor=db.cursor()
cursor.execute("INSERT INTO cissql (date,cis,alt,temp1)\nVALUES ('%s','%s','%s','%s')"%(xt,c,d,e));
db.commit()
db.close()
```

Listing kodu 3. Wysłanie pomiarów do bazy danych czujnika BMP180.

Komendą „*print()*” wyświetlany zostaje w oknie terminala wynik pomiaru, który w dalszej części kodu zostaje przesłany do bazy danych. Polecenie „*print()*” w

tym przypadku służy tylko do weryfikacji czy zapisane do bazy danych pomiary są zgodne z pojawiającymi się w terminalu. Niezbędną informacją poza samymi pomiarami jest przybliżona data oraz godzina ich wykonania. Do realizacji tego celu zastosowane zostało polecenie „`xt=time.strftime('%Y-%m-%d %H:%M:%S')`”. Po wykonaniu tego wiersza do zmiennej „`xt`” zostaje zapisana data w formacie rok-miesiąc-dzień godzina-minuta-sekunda. Wiersz kodu „`cursor=db.cursor()`” umożliwia wykonywanie poleceń programu w otwartej sesji bazy danych. Następnym krokiem jest wpisanie do stworzonej wcześniej tabeli wszystkich danych przy użyciu komendy „`cursor.execute()`”. W miejsce wykonania należy umieścić odpowiednie polecenie w języku MySQL. W tym przypadku jest to „`INSERT INTO cissql (date,cis,alt,temp1)\nVALUES ('%s','%s','%s','%s')"%(xt,c,d,e)`”, które należy interpretować następująco: do tabeli o nazwie „`cissql`” w kolumny o nazwach „`date`”, „`cis`”, „`alt`” oraz „`temp1`” wstaw wartości „`xt`”, „`c`”, „`d`”, „`e`”, które w tym przypadku są datą i wynikami pomiarów. By potwierdzić polecenie w bazie danych stosuje się „`db.commit()`”, natomiast do jej zamknięcia „`db.close()`”.

## 5.2DHT21 (AM2301)

Skrypt dla czujnika temperatury i wilgotności jest podobny w strukturze do czujnika ciśnienia, dlatego przedstawione poniżej zostaną tylko najważniejsze informacje. Połączenie z bazą danych i deklaracja bibliotek zaprezentowana w listingu kodu 4.

```
import MySQLdb
import subprocess
import time
import datetime
import linecache
import math
import decimal
import os,sys
import numpy as np
db=MySQLdb.connect("mysql.agh.edu.pl","pzwolak","xxxxxxxxxxxxxxxx","pzwolak")
```

Listing kodu 4. Sposób importu bibliotek odpowiadających m.in. za funkcje matematyczne, obsługę czasu czy uruchomienie podprogramu oraz połączenie z bazą danych.

Biblioteka „`subprocess`” została zaimportowana w celu umożliwienia otwarcia przez skrypt podprogramu napisanego w języku „C”. Po uruchomieniu podprogramu do pliku tekstowego zostaje zapisany pomiar z czujnika natężenia światła. Kolejną

biblioteką konieczną do czytania informacji z pliku jest „linecache”. Do otwierania oraz odczytu treści z pliku wykorzystano bibliotekę „os” oraz „sys”. Biblioteki „math” oraz „decimal” pozwalają na dostęp do funkcji matematycznych. Listing 5 przedstawia uruchomienie podprogramu oraz otwarcie, odczyt i zamknięcie pliku.

```
while licz<6:
    p=subprocess.Popen('/home/pi/Desktop/TIW/./a.out')
    time.sleep(10)
    plik=open('tekst.txt','r')
    try:
        t=plik.readline()
        w=plik.readline()
        while t=="":
            print('Błąd odczytu, plik nie posiada zawartosci')
            t=cc
        while w=="":
            print('Błąd odczytu, plik nie posiada zawartosci ')
            w=dd
        cc=t
        dd=w
        t=float(t)
        w=float(w)
    finally:
        plik.close()
    cc=t
    dd=w
    tab.append(t)
    tabb.append(w)
    licz+=1
    time.sleep(40)
    print('temp',t)
    print('wilg',w)
```

Listing kodu 5. Uruchamianie podprogramu dla czujnika DHT21, otwarcie pliku, odczyt pomiarów temperatury i wilgotności, zamknięcie pliku.

Powyższy kod z listingu 5 wykonywany jest podobnie jak w przypadku czujnika ciśnienia czyli 6 pomiarów co 50 sekund. W celu otrzymania wyniku niezbędne jest uruchomienie podprogramu. Takie działanie możliwe jest dzięki poleceniu „*p=subprocess.Popen('/home/pi/Desktop/TIW/./a.out')*”, które uruchamia podprogram o nazwie „*a.out*” znajdujący się w „*/home/pi/Desktop/TIW/*”. Po uruchomieniu podprogram zapisuje pomiary w pliku tekstowym. Do otwarcia pliku z pomiarami użyta została komenda „*plik=open('tekst.txt','r')*”, gdzie ‘*r*’ oznacza tylko do odczytu, natomiast do odczytu zawartości pliku służy funkcja „*plik.readline()*”. Z niewiadomych

przyczyn podprogram do pliku tekstowego średnio raz na dwa dni zapisuje pusty plik. Powoduje to wystąpienie błędu w trakcie odczytu i zamknięcie się programu. Rozwiązaniem tego problemu jest użycie pętli sprawdzającej „*while* *t*==“:” oraz „*while* *w*==“:” czy zapisane dane do zmiennych „*w*” oraz „*t*” nie są puste, jeśli sprawdzony warunek będzie prawdziwy to następuje przypisanie do zmiennych wartości z poprzedniego pomiaru oraz wypisaniu na konsoli błędu. Takie rozwiązanie nie wpłynie negatywnie na ogólny wynik ze względu na rzadkość występowania oraz czas pomiędzy pomiarami wynoszący 50 sec. W dalszej części informacje zapisywane są do bazy danych tak jak w przypadku czujnika BMP180 tylko do innej tabeli.

### 5.3BH1750

Kolejnym czujnikiem, który wymaga oprogramowania do sprawnego wykonania pomiarów jest czujnik natężenia światła. Poniżej zostanie przedstawiony oraz zinterpretowany skrypt, natomiast szczególna uwaga zostanie zwrócona na możliwość dokładności pomiaru tego czujnika. Połączenie z bazą danych oraz deklaracja właściwych bibliotek dla czujnika BH1750 przedstawiona w listingu kodu 6.

```
import MySQLdb
import sys
import time
import smbus
import datetime
db = MySQLdb.connect("mysql.agh.edu.pl","pzwolak","xxxxxxxxxxxxxxxx","pzwolak")
```

Listing kodu 6. Importowanie bibliotek sterownika czujnika BH1750 oraz połączenie z bazą danych.

Większość bibliotek została opisana już we wcześniejszych podrozdziałach dlatego opisana zostanie tylko biblioteka „*smbus*”. SMBus to skrót od System Management Bus, który można przetłumaczyć jako Magistrała Zarządzania Systemem. Biblioteka ta umożliwia odwołanie się do protokołu I2C, który jest szeregową dwukierunkową magistralą umożliwiającą przesyłanie w urządzeniach elektronicznych informacji. Kolejnymi niezbędnymi liniami kodu są deklaracje adresu I2C oraz dokładności pomiaru (listing kodu 7).

```

DEVICE = 0x23
POWER_DOWN = 0x00
POWER_ON = 0x01
RESET = 0x07

CONTINUOUS_LOW_RES_MODE = 0x13
CONTINUOUS_HIGH_RES_MODE_1 = 0x10
CONTINUOUS_HIGH_RES_MODE_2 = 0x11
ONE_TIME_HIGH_RES_MODE_1 = 0x20
ONE_TIME_HIGH_RES_MODE_2 = 0x21
ONE_TIME_LOW_RES_MODE = 0x23

bus = smbus.SMBus(1)

```

Listing kodu 7. Deklaracja adresu czujnika BH1750 oraz dokładności jego pomiaru.

Pierwsza linia kodu listingu 7 służy do zadeklarowania zmiennej adresu podłączonego urządzenia czyli „*DEVICE = 0x23*”. 0x23 jest domyślnym adresem dla tego czujnika. Następne linie kodu należy interpretować jako:

- wyłączenie aktywności – „*POWER\_DOWN = 0x00*”, które przyczynia się do obniżenia zużycia prądu
- włączenie aktywności – „*POWER\_ON = 0x01*”
- reset odczytanych wartości – „*RESET = 0x07*”

Następnie należy zdefiniować dokładność pomiarową. W tym przypadku mamy do wyboru 6 rodzajów dokładności i można je interpretować następująco:

- Pomiar o dokładności 4 lx trwający około 16ms  
„*CONTINUOUS\_LOW\_RES\_MODE = 0x13*”
- Pomiar o dokładności 1 lx trwający około 120ms  
„*CONTINUOUS\_HIGH\_RES\_MODE\_1 = 0x10*”
- Pomiar o dokładności 0.5 lx trwający około 120ms  
„*CONTINUOUS\_HIGH\_RES\_MODE\_2 = 0x11*”
- Pomiar o dokładności 1 lx trwający około 120ms z automatycznym wyłączeniem aktywności „*ONE\_TIME\_HIGH\_RES\_MODE\_1 = 0x20*”
- Pomiar o dokładności 0.5 lx trwający około 120ms z automatycznym wyłączeniem aktywności „*ONE\_TIME\_HIGH\_RES\_MODE\_2 = 0x21*”
- Pomiar o dokładności 4 lx trwający około 16ms z automatycznym wyłączeniem aktywności „*ONE\_TIME\_LOW\_RES\_MODE = 0x23*”

Podsumowując dokładność pomiarową BH1750 należy podkreślić, iż im dokładność większa tym czas odpowiedzi o pomiarze jest dłuższy oraz zużycie energii większe. Przy zasilaniu z sieci rekomendowane jest użycie największej dokładności o

ile taka jest wymagana. W tym projekcie używane jest mieszane źródło zasilania, z tego powodu wybrana została dokładność pomiarowa rzędu 1 lx z automatycznym wyłączeniem aktywności.

Kolejną kwestią jest konwertowanie wyniku binarnego na dziesiętny, którego kod dostępny jest w listingu kodu 8.

```
def convertToNumber(data):  
    return ((data[1] + (256 * data[0])) / 1.2)  
  
def readLight(addr=DEVICE):  
    data = bus.read_i2c_block_data(addr, ONE_TIME_HIGH_RES_MODE_1)  
    return convertToNumber(data)
```

Listing kodu 8. Funkcja pobierająca pomiar z czujnika BH1750 oraz konwertująca go z systemu binarnego na dziesiętny.

By dokonać pomiaru należy wywołać funkcję „*readLight()*” w której przesyłanym parametrem jest adres czujnika w protokole I2C. Wewnątrz tej funkcji znajduje się wiersz służący do odczytu wyniku przy użyciu biblioteki SMBus, w którym jako parametry przesyłany jest adres urządzenia oraz wybrana przez nas dokładność pomiarowa. Następnie zwracaną wartością funkcji jest wynik w postaci liczby dziesiętnej, która została przekonwertowana dzięki wspomnianej wcześniej funkcji. Następne kroki wyglądają praktycznie tak samo jak w przypadku wcześniej opisanych czujników. Pobierane jest kilka pomiarów przy użyciu funkcji „*readLight()*” z których wyliczana średnia przesyłana zostaje do bazy danych.



## 6. Oprogramowanie strony internetowej

Opracowane w rozdziale piątym oprogramowanie czujników umożliwiło zapis pomiarów do bazy danych MySQL, które można z powodzeniem zobaczyć poprzez zalogowanie się do niej. Dokonać tego może jedynie osoba uprawniona, posiadająca login oraz hasło do bazy. Ze względu na to, iż ważnym założeniem pracy jest możliwość prezentacji wyników na stronie internetowej konieczne jest skorzystanie z technologii takich jak HTML, JavaScript oraz PHP. Technologie te umożliwiły zbudowanie prostej, przejrzystej strony internetowej, do której dostęp może mieć każdy zainteresowany użytkownik.

### 6.1 Praca po stronie serwera

Prawidłowy odczyt wyników z bazy danych możliwy jest dzięki zastosowaniu technologii PHP, która między innymi zostanie przedstawiona i opisana w tym podrozdziale na przykładzie czujnika natężenia światła. Na wstępie należy zaznaczyć, iż wyjściowe dane pomiarów zapisywane są w postaci tekstu, zabieg ten jest konieczny by móc przekazać informacje do dalszego przetwarzania przez asynchroniczny JavaScript i XML czyli w skrócie AJAX. Dane wyjściowe zapisywane są więc w postaci tekstowej z notacją obiektów JS dzięki wykorzystaniu formatu danych JSON [10] co w rozwinięciu oznacza **JavaScript Object Notation**. Takie rozwiązanie pozwala na wymianę danych między przeglądarką a serwerem bez konieczności odświeżania całej strony. Poniżej przedstawione zostało w listingu 9 definiowanie bazy danych oraz struktury strony jako typ `application/json`.

```
header('Content-Type: application/json');

define('DB_HOST', 'mysql.agh.edu.pl');
define('DB_USERNAME', 'pzwolak');
define('DB_PASSWORD', 'xxxxxxxxxxxxxxxx');
define('DB_NAME', 'pzwolak');

$mysqli = new mysqli(DB_HOST, DB_USERNAME, DB_PASSWORD, DB_NAME);

if(!$mysqli)
{
    die("Connection failed: " . $mysqli->error);
}
```

Listing kodu 9. Definiowanie typu zawartości strony jako `application/json` oraz połączenie z bazą danych.

Poleceniami „*define*” zdefiniowane zostają informacje o dostępie do bazy danych, które wykorzystane są do stworzenia połączenia i przypisania go do zmiennej „*\$mysqli*”. Jeśli dane zostały poprawnie wprowadzone oraz nie występują żadne błędy po stronie serwera, połączenie utworzy się. Instrukcja warunkowa „*if(!\$mysqli)*” sprawdza czy połączenie się powiodło, gdyby wystąpiły problemy zostanie wyświetlony błąd połączenia. Kiedy połączenie z bazą danych przebiegło pomyślnie następnym krokiem jest wyciągnięcie z niej odpowiednich informacji a w szczególności chodzi o odpowiedni dzień, tydzień lub miesiąc, który użytkownik poda lub nie. Listing 10 przedstawia wykorzystanie informacji z pliku, które wpisał użytkownik w celu otrzymania oczekiwanego wykresu.

```
$plik_nazwa = 'dane.txt';  
$wynik = fopen($plik_nazwa, 'r+');  
$wynikgotowy = fread($wynik, filesize($plik_nazwa));  
file_put_contents('dane.txt', '1');  
fclose($wynik);
```

Listing kodu 10. Odczyt informacji podanych przez użytkownika.

Linie kodu przedstawione na listingu kodu 10 wczytują informacje do zmiennej „*\$wynikgotowy*” oraz zmieniają zawartość pliku na „1” by po odświeżeniu strony wyświetlone zostały domyślne wyniki z ubiegłego dnia, tygodnia lub miesiąca. Następnym zadaniem jest wykonanie obliczeń czasu w celu otrzymania odpowiednich wyników z bazy danych. Poniżej, w listingu 11 zaprezentowane zostały procedury uzyskania żądanych wyników.

```
$czas = time();  
$czass = time();  
$z1=0;  
$z2=0;  
$z1=86400*$wynikgotowy;  
$z2=86400*$wynikgotowy+86400;  
$czas = $czas - $z1;  
$czass = $czass - $z2;  
$czas = date('Y-m-d', $czas);  
$czass = date('Y-m-d', $czass);  
  
$luux="SELECT lux, date FROM luxsql WHERE date BETWEEN '{$czass}' AND '{$czas}'";  
$stabela=mysqli_query($mysqli, $luux);
```

Listing kodu 11. Obliczenia zakresu czasu do wyszukania w bazie danych oraz pobieranie na jego podstawie oczekiwanych wyników z określonego przedziału czasowego.

Zawartość powyższego listingu przedstawia pobieranie aktualnego czasu i zapisanie go do zmiennych. Następnie obliczane są zmienne zakresu dla wybranego przez użytkownika dnia. „ $\$z1=86400*\$wynikgotowy;$ ” – 86400 jest ilością sekund jednego dnia, wynik ten jest mnożony przez dane pobrane od użytkownika. W zmiennej „ $\$z2$ ” wynik jest taki jak w przypadku „ $\$z1$ ” + 86400 sekund. Takie zastosowanie pozwala na stworzenie zakresu jednego dnia. Następnie zakres ten jest odejmowany od aktualnego czasu oraz ustawiany jest format daty na rok-miesiąc-dzień. Format ustawiony w ten sposób umożliwia ustawienie zakresu pełnego dnia od godziny 0:00 do 0:00. Kolejną czynnością jest wyfiltrowanie z tablicy, właściwych danych dzięki zastosowaniu utworzonego wcześniej zakresu dnia, które w tym przypadku zapisywane są do zmiennej „ $\$luux$ ”. Ostatni wiersz listingu 11 odpowiada za pobieranie i zapis do zmiennej  $\$tabela$  wszystkich informacji ( w tym przypadku o natężeniu światła ) wyfiltrowane uprzednio i przypisane przy pomocy zmiennej „ $\$luux$ ”. Kolejnym krokiem jest zapisanie wszystkich wcześniej wybranych pomiarów do tablicy w celu późniejszego wyliczenia średniej arytmetycznej z pomiarów. Pętla przypisująca wyniki do konkretnych tablic przedstawiona została w listingu 12.

```
$licznik=0;
while($row=mysqli_fetch_assoc($tabela))
{
    $natezenie[] = (float)$row['lux'];
    $dzis[] = $row['date'];
    $licznik=$licznik+1;
}
$licznik2=$licznik/25;
```

Listing kodu 12. Pętla przypisująca pomiary do konkretnych tablic.

Po wykonaniu powyższych poleceń niezbędną czynnością jest uporządkowanie wszystkich pomiarów. Jeśli co 5 minut do bazy danych zapisywany jest pomiar, daje to 12 wyników na godzinę czyli 288 na dobę. Taka ilość liczb wyświetlona na stronie z pewnością odstraszyłaby każdego zainteresowanego użytkownika, dlatego konieczne jest obliczenie średniej arytmetycznej wyników dla każdej godziny co przekłada się na zapisanie do tabeli końcowej 25 wyników. Zawartość skryptu umożliwiająca wykonanie takich obliczeń przedstawiona jest poniżej w listingu 13.

```

$a=0;
$suma=0;
$suma2=0;
$k=0;
$wynik=0;
$data2 = array();
for($i=0; $i<25;$i++)
{
    for($j=0; $j<$licznik2; $j++)
    {
        $a=$natezenie[$k+$j];
        $suma=$suma+1;
        $suma2=$suma2+$a;
    }
    $k=$k+$licznik2;
    $wynik=$suma2/$suma;
    $data2[$i]=$wynik;
    $a=0;
    $suma=0;
    $suma2=0;
}

```

Listing kodu 13. Pętle obliczające średnią arytmetyczną pomiarów natężenia światła oraz ich zapis do tablicy wyjściowej.

Gotowe, wyliczone i wyfiltrowane pomiary natężenia światła zapisane w tablicy „*\$data2[]*” należy wyświetlić poleceniem „*print json\_encode(\$data2);*” w celu dalszej ich interpretacji przez JavaScript. Na chwilę obecną wyniki przedstawione są w postaci samych liczb, które nie są dostępne z pozycji strony internetowej natomiast można je zobaczyć podając w przeglądarce ścieżkę do skryptu. W tym przypadku url to: <http://student.agh.edu.pl/~pzwolak/dziala.php> . Po uruchomieniu powyższego adresu strony wyświetlą się wyniki jednego przykładowego dnia pomiarów, które przedstawiono na rysunku 5.

JSON	Nieprzetworzone dane
Zapisz	Kopiuuj
0:	0
1:	0
2:	0
3:	0
4:	0
5:	0
6:	0
7:	3.65
8:	80.183333333333
9:	254.68333333333
10:	554.025
11:	993.56666666667
12:	654.75
⋮	
24:	0.49166666666667

Rysunek 5. Wyniki natężenia światła wygenerowane w trakcie jednej doby.

## 6.2 Tworzenie wykresów

Nieodłącznym elementem każdej strony jest jej oprawa graficzna. Stworzenie strony internetowej bez atrakcyjnych sliderów, grafik czy interakcji z użytkownikiem nie zachęca do korzystania z jej zawartości. Istnieją także skomplikowane strony, które w głównej mierze mają umożliwić użytkownikowi przedstawienie różnych informacji w jasny i klarowny sposób, przykładem może być <https://www.google.com/analytics/>. Ważnym celem tej pracy jest właściwe przedstawienie wyników w czytelny sposób dla użytkownika. W realizacji tego postanowienia konieczne było posłużenie się takimi technologiami jak JavaScript z wykorzystaniem biblioteki charts.js umożliwiającą np. rysowanie wykresu z otrzymanych i przetworzonych wcześniej wyników. Poniżej w listingu 14 przedstawiona jest część kodu napisana w języku JavaScript odpowiedzialna za tworzenie się wykresu.

```

var wartosci = []
wartosci.push(data[0]);
.....
wartosci.push(data[24]);
var chartdata =
{
  labels: ["00:00", "01:00", "02:00", ..... "23:00", "24:00"],
  datasets :
  [
    {
      label: "Natężenie światła [lx]",
      borderColor : 'rgba(70,70,60, 0.7)',
      pointBorderColor : 'rgba(70,70,60, 1)',
      borderWidth : 5,
      fill: true,
      backgroundColor : 'rgba(70,70,60, 0.2)',
      .....
      pointHoverBackgroundColor : 'rgba(90,255,255,1)',
      pointHoverBorderColor : 'rgba(70,115,87, 1)',
      data: wartosci
    }
  ]
};

var ctx = $("#mycanvas8");

var barGraph = new Chart(ctx,
{
  type: 'line',
  data: chartdata
});

```

Listing kodu 14. Część kodu odpowiedzialna za budowę wykresu.

Zawartość kodu znajdującego się w listingu 14 można interpretować następująco: na początku tworzona jest tablica do której przypisywane są wartości otrzymane jako wynik wyjściowy ze skryptu opisanego w podrozdziale 6.1, który w dalszej części zostanie przekazany jako zmienna danych do stworzenia wykresu. W dalszej części do zmiennej „*chartdata*” przypisywane są informacje o wykresie oraz jego ustawieniach takie jak:

- nazwa tabeli: „*label: "Natężenie światła [lx]"*,”
- kolor tła tabeli: „*borderColor : 'rgba(70,70,60, 0.7)'*,”
- nazwy na osi x: „*labels: ["00:00", "01:00", "02:00", ..... "23:00", "24:00"]*,”
- kolor punktu: „*pointBorderColor : 'rgba(70,70,60, 1)'*,”
- kolor tła punktu: „*pointHoverBackgroundColor : 'rgba(90,255,255,1)'*,”

- dane pomiarowe: „*data: wartości*” itd.

W dalszej części ustawiana jest nazwa konieczna do narysowania wykresu oraz typ, który w tym przypadku jest liniowy.

### 6.3 Strona internetowa

Strona internetowa tego projektu jest ogólnodostępna pod adresem: <http://student.agh.edu.pl/~pzwolak/index.php>. By móc zobaczyć wyniki należy zalogować się wpisując „uzytkownik” jako login i hasło. Strona została stworzona w zgodności z najnowszą technologią HTML 5 oraz została wyposażona w kilka nowych znaczników semantycznych umożliwiających łatwiejsze rozpoznanie kodu robotom analizującym stronę. Zastosowanie technologii PHP pozwoliło na użycie sesji logowania na stronie. Dlatego konieczne było umieszczenie już na samym początku kodu strony skrypt sprawdzający zgodność sesji. Dopiero po tej części znajduje się kod HTML odpowiedzialny za metadane oraz wygląd strony. W części „<head>” niezbędne było zastosowanie takich znaczników jak:

- <meta charset="utf-8"> - odpowiedzialne za prawidłowe kodowanie znaków w standardzie UTF-8
- <title></title> - tytuł strony
- <meta name="description" content="... "> - opis strony
- <meta name="keywords" content="..."> - słowa kluczowe
- <meta name="author" content="Patryk Zwolak"> - autor strony
- <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1"> - wymuszenie na przeglądarce Internet Explorer użycie najnowszej wersji silnika
- <link rel="stylesheet" href="main.css"> - odwołanie do arkusza stylów

Po omówieniu sekcji „head” niezbędne jest przedstawienie części zawartości sekcji „body”. Na samym początku tej sekcji konieczne jest wywołanie funkcji odwołującej się do JavaScript. W przypadku strony głównej przedstawiającej aktualne pomiary z czujników taki kod będzie wyglądał następująco, tak jak w listingu 15.

```
<body onload="wykres1(); wykres2(); wykres3(); wykres4(); wykres5(); wykres6(); aktualne();">
```

Listing kodu 15. Deklaracja rozpoczęcia sekcji body oraz wywołanie funkcji JavaScript.

W dalszej części znajduje się znacznik „<header>”, w którym umieszczony został nagłówek pracy inżynierskiej oraz menu, które także zostało umieszczone w nowym znaczniku symantycznym „<nav>”. Znacznik ten może być używany przez syntetyzator mowy, co umożliwi korzystanie z menu osobom niedowidzącym. Kolejną konieczną częścią do omówienia jest zawartość odpowiedzialna za pojawienie się wykresu na stronie. Przykładowy kod odpowiadający za tą czynność przedstawiony został w listingu 16.

```
<div class="chart-container">
  <canvas id="mycanvas6"></canvas>
  <script type="text/javascript" src="jquery.min.js"></script>
  <script type="text/javascript" src="Chart.js"></script>
  <script type="text/javascript" src="app6.js"></script></br></br>
</div>
```

Listing kodu 16. Kod odpowiedzialny za pojawienie się wykresu na stronie.

W powyższym listingu na początku definiowany jest kontener w którym będzie się znajdował wykres. Jego dokładne ustawienia szerokości i wysokości opisane są w arkuszu stylu, do którego odwołuje się poprzez klasę. Znacznik „<canvas>” pozwala na wyświetlenie skryptowej zawartości w ustalonym przez „<div>” miejscu. W tym przypadku jest to wyświetlenie wykresu wilgotności. W kolejnej części kodu umieszczany jest skrypt, „text/javascript” oznaczający, iż typ pliku do którego skrypt ma się odwołać, napisany jest tekstem w języku JavaScript. W miejscu „src=” podana jest ścieżka z nazwą do pliku. Do poprawnego działania wykresu konieczne jest użycie:

- „jquery.min.js” – biblioteka upraszczająca programowanie w JavaScript oraz zwiększająca jego możliwości
- „Chart.js” – biblioteka zawierająca niezbędne informacje, które umożliwiają tworzenie wykresów
- „app6.js” – skrypt pobierający dane pomiarowe oraz ustawienia wykresu m.in. kolor, kolor linii, wielkość punktu itp.

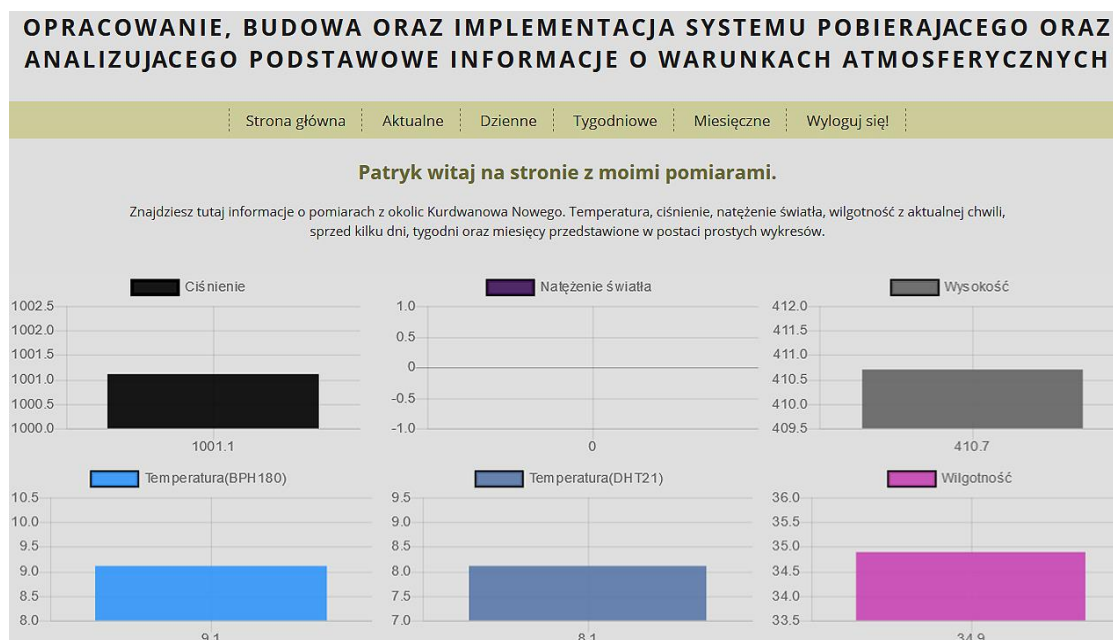


## 7. Testy

System pobierający oraz analizujący warunki atmosferyczne może pracować zdalnie lub stacjonarnie. Do pracy zdalnej konieczne jest połączenie się z siecią Wi-Fi. Pierwszy test polega na sprawdzeniu połączenia między urządzeniem (Raspberry Pi 3) a punktem dostępu. Do pierwszego połączenia z punktem dostępu ważne jest by podać hasło dostępu. Przy kolejnej próbie Raspberry połączy się automatycznie z siecią. Do sprawdzenia prawidłowości połączenia można posłużyć się programem Advanced IP Scanner na PC z systemem Windows lub Fing - Network Tools na urządzenia mobilne z systemem android.

Kolejnym testem było sprawdzenie czy po prawidłowym połączeniu z punktem dostępu urządzenie wysyła informacje do bazy danych. Jeśli wszystko przebiegło pomyślnie w bazie danych pojawią się wyniki z aktualną datą i czasem.

Ostatnim testem jest sprawdzenie działania strony internetowej, która dostępna jest pod adresem: <http://student.agh.edu.pl/~pzwolak/wyniki.php>. Rysunek 6 przedstawia zrzut ekranu strony internetowej z aktualnymi pomiarami.



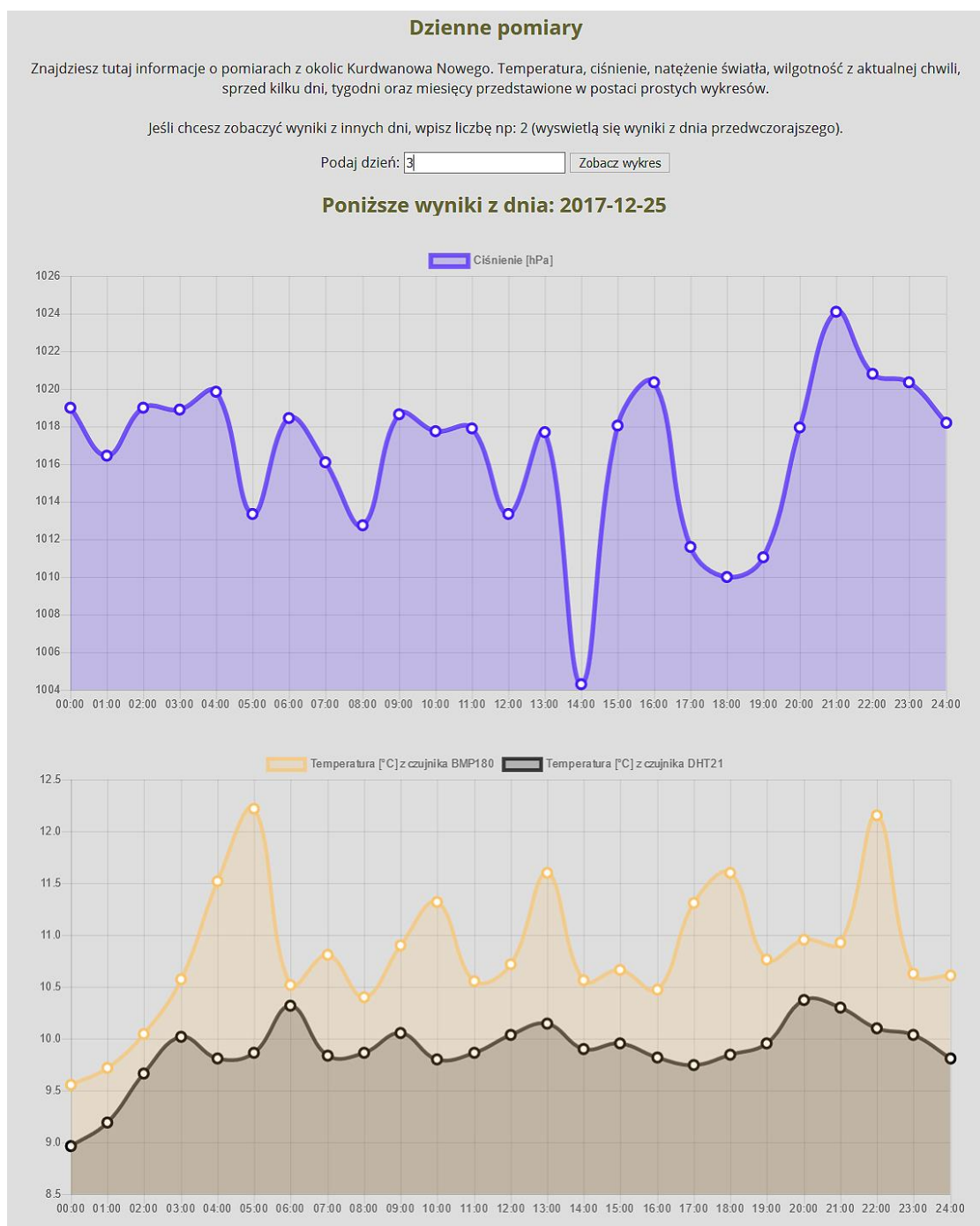
Rysunek 6. Zrzut ekranu strony internetowej z aktualnymi pomiarami.

Powyższe zdjęcie przedstawia zrzut ekranu strony internetowej, na której prezentowane są aktualne wyniki pomiarów. Przejście do wyżej wymienionego adresu wyświetli stronę, na której pojawiają się aktualne wyniki odświeżane co 5 sekund. Do sprawdzenia

prawidłowości pomiarów posłużyłem się stroną internetową <https://pogoda.onet.pl/prognoza-pogody/podgorze-duchackie-392088> oraz <https://pogoda.interia.pl/prognoza-szczegolowa-krakow-podgorze-duchackie,cId,2838652> do których porównywane były wyniki z pracy. Otrzymane wyniki pomiarowe różnią się od tych na stronie dla:

- Ciśnienia: zaniżone - 1-2 hPa,
- Temperatura (BPH180): zawyżone - 2-3°C,
- Temperatura (DHT21): zawyżone - 1-2°C,
- Wilgotność: zaniżone - 10-20%.

Powyższe różnice prawdopodobnie wynikają z umiejscowienia (zabudowany balkon) systemu pomiarowego w porównaniu do stacji meteorologicznych powyższych stron internetowych, dlatego wyniki mogą od nich odbiegać. Ponadto wykonane testy termometrem cieczowym wykazały, iż różnica temperatury dla czujnika BPH180 jest zawyżana o około 1°C. Czujnik DHT21 i termometr cieczowy wykazywały taką samą wartość. Higrometr wykazał procentową różnicę pomiarów o około 5%. Ważną częścią testu jest sprawdzenie poprawności filtrowania wyników pomiarów dla np. całego dnia (Rysunek 7).



Rysunek 7. Zrzut ekranu wyników pomiarów z dnia 25-12-2017r.

Powyższy rysunek 7 ilustruje wyniki pomiarów (ciśnienia i temperatury) z dnia 25-12-2017r. Po podaniu odpowiedniej wartości (w tym przypadku 3) system pozyskuje z bazy danych informacje o pomiarach, które przedstawione są w postaci wykresu.

## 8. Zakończenie

Głównym celem pracy inżynierskiej było opracowanie, budowa oraz implementacja systemu pobierającego oraz analizującego warunki atmosferyczne. W pracy udało się zrealizować wszystkie założenia oraz cele szczegółowe.

W powyższych rozdziałach pracy inżynierskiej opisane zostały szczegółowo wszystkie podzespoły, bez których system nie mógł by funkcjonować. Przedstawiona została charakterystyka mikrokontrolera oraz czujników wraz ze schematami połączeń. Dalsza część pracy zawiera omówienie większości zastosowanych technologii do realizacji tego projektu. Przedstawione zostały podstawy każdej technologii poczynając od historii aż po praktyczne zastosowanie i omówienie podstawowych poleceń.

Oprogramowanie czujników przedstawione w postaci części kodu zostało wytłumaczone oraz zinterpretowane w sposób taki by osoba, która na co dzień nie ma do czynienia z kodem była w stanie zrozumieć zasadę działania opracowanego w ramach niniejszej pracy systemu.

W pracy przedstawiono działanie strony internetowej od strony technicznej, pracę po stronie serwera, tworzenie się wykresów oraz sam szkielet strony internetowej.

System pobierający oraz analizujący warunki atmosferyczne został zrealizowany z powodzeniem. W trakcie tworzenia projektu zauważyłem jego duży potencjał w perspektywie jego dalszego rozwoju i rozbudowy. Dostęp różnorodnych czujników oraz wydajność mikrokontrolera umożliwiają jego rozbudowę o takie czujniki jak czujnik opadów deszczu, czujnik kierunku oraz prędkości wiatru czy czujnik pyłu. Ponadto stworzenie odpowiedniego oprogramowania może umożliwić wyświetlenie np. natężenia światła jako odpowiednie zachmurzenie lub jego brak w postaci graficznej. Niniejszą pracę można również potraktować jako instrukcja budowania domowej stacji meteorologicznej.

## 9. Literatura

- [1] <https://kamami.pl/komputery-raspberry-pi-3/561006-raspberry-pi-3-model-b-komputer-z-bcm2837-i-1gb-ram.html>
- [2] <https://www.raspberrypi.org/downloads/>
- [3] <https://botland.com.pl/czujniki-temperatury/1884-czujnik-temperatury-i-wilgotnosci-dht21-am2301-w-obudowie.html>
- [4] <https://www.sparkfun.com/products/retired/11824>
- [5] <http://www.instructables.com/id/BH1750-Digital-Light-Sensor/>
- [6] M. Lutz *Wprowadzenie Python*, wydanie IV, Wydawnictwo Helion 2011
- [7] W. J. Gilmore *PHP i MySQL od podstaw*, wydanie IV, Wydawnictwo Helion
- [8] J. Duckett *HTML i CSS*, Wydawnictwo Helion
- [9] D. Sawyer McFarland *JavaScript i jQuery, nieoficjalny podręcznik*, wydanie III, Wydawnictwo Helion 2015
- [10] [https://www.w3schools.com/js/js\\_json\\_intro.asp](https://www.w3schools.com/js/js_json_intro.asp)