

Wrocław, 11 czerwca 2019

Emilia Pawłaszek, 241279
Łukasz Szumilas, 236068

PN TN 13:15

Sprawozdanie z projektu przedmiotu „Architektura Komputerów 2”

Rok akad. 2018/2019, kierunek: INF

Prowadzący:
dr inż. Dominik Żelazny

Spis treści

1	Założenia projektu	3
2	Wykorzystane narzędzia	3
3	Realizacja	3
3.1	Czujnik DHT11	3
3.2	Czujnik BMP180	4
3.3	Czujnik BH1750	5
3.4	Moduł WiFi ESP8266	5
4	Programowanie	6
5	Schematy połączeń	7
6	Trudności w realizacji projektu	8
7	Zdjęcia systemu	8
8	Wnioski	10
	Bibliografia	10

1 Założenia projektu

Celem projektu jest stworzenie systemu pobierającego dane pogodowe i przesyłającego ich na urządzenie odbierające dane. System bez ingerencji człowieka ma być w stanie mierzyć temperaturę, wilgotność powietrza, ciśnienie atmosferyczne i natężenie światła.

Dodatkowo za pomocą modułu wifi i przeglądarki układ łączy się z odbiorcą i automatycznie przesyła dane.

2 Wykorzystane narzędzia

System powstał w oparciu o model Arduino Uno, wraz z rozszerzeniami:

- DHT11 – czujnik wilgotności i temperatury,
- BH1750 – czujnik natężenia światła,
- BMP180 – czujnik ciśnienia i temperatury,
- ESP8266 – moduł wifi.

Dodatkowo:

- Przewód USB
- Płytki stykowe
- Rezystory
- Przewody, kable

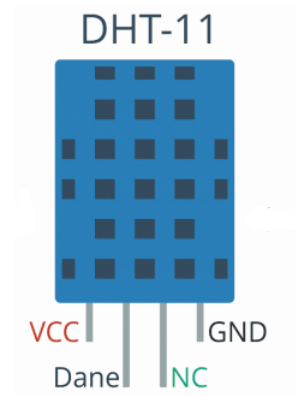
3 Realizacja

Realizacja rozpoczęła się od poznania teoretycznych zagadnień dotyczących czujników: ich wejść, wyjść, sposobu działania i przesyłu danych na platformę Arduino.

3.1 Czujnik DHT11

Czujnik ten odpowiada za pokazywanie pomiarów wilgotności względnej. Wartości te są z przedziału 0-1 i nie posiadają jednostki, więc w projekcie grupa przedstawiała je jako procent ciśnienia cząstkowego pary wodnej zawartej w powietrzu do ciśnienia nasycenia nad płaską powierzchnią czystej wody, gdzie: 0% to powietrze całkowicie suche, 100% maksymalnie wilgotne. Sensor ten posiada także wbudowany termometr w zakresie 0-50 stopni Celsjusza, jednak nie został on użyty w projekcie.

Czujnik ten składa się z następujących wejść:

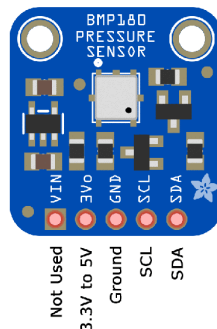


Rysunek 1: Czujnik DHT11 i jego wejścia

VCC podpiną się do zasilania 5V, **GND** oznacza Ground. Jest to masa, która służy do wyrównania potencjałów w układzie. Drugi pin z danymi był podłączony do jednego wejścia na Arduino *PWM* (Pulse Width Modulation), co oznacza modulację szerokości impulsu. Z tak pobieranych danych jest mierzony procent czasu, przez jaki sygnał ma potencjał wysoki i kiedy rzeczywiście informacje o wilgotności są przekazywane na mikrokontroler. Oprócz tego przy wyprowadzaniu danych trzeba skorzystać z rezystora pull-up, by podtrzymać (kiedy zachodzi taka potrzeba) stan wysoki i zapobiec jednocześnie stanom nieustalonym.

3.2 Czujnik BMP180

W przypadku tego urządzenia powinno się zwrócić szczególną uwagę na wejście **VCC**, które podpiną się do 3.3V, gdyż tylko takie napięcie może przyjąć.



Rysunek 2: Czujnik BMP180

Po drugiej stronie widać wejście **GND**, zewnętrzne piny odpowiadają więc za to samo co piny o tej samej nazwie opisane w *DHT11*. Najważniejsze to **SCL** i **SDA**. Należą one do dwukierunkowej, szeregowej magistrali *I²C*, gdzie SCL odpowiada to linia zegara natomiast SDA to linia danych. Obie są na stałe podciągnięte do źródła zasilania dzięki wbudowanemu rezystorowi pull-up, przez co nie ma potrzeby dodawania do schematu kolejnego opornika. Na płytce Arduino znajdują się specjalne wejścia do komunikacji z taką magistralą o takich samych nazwach.

BMP180 charakteryzuje się pomiarem ciśnienia w zakresie od 300 do 1100 hPa, co daje możliwość określenia wysokości od +9000 do -500 metrów względem poziomu morza.

Czujnik, by pokazać poprawne pomiary ciśnienia, musi w pierw pobrać aktualną temperaturę, grupa więc zdecydowała na pomiar stopni Celsjusza tym sensorem zamiast *DHT11*. W

programie można także obliczać aktualną wysokość nad poziomem morza dzięki wbudowanemu algorytmowi, mając na wejście informacje o ciśnieniu nad poziomem morza.

3.3 Czujnik BH1750

BH1750 to cyfrowy czujnik natężenia światła działający w zakresie 1 - 65535 lx (luks), gdzie luks określany jest jako oświetlenie wywołane przez równomiernie rozłożony strumień świetlny o wartości równej 1 lumen (lm) padający na powierzchnię $1m^2$.

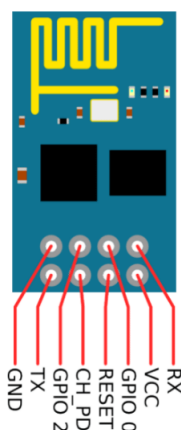


Rysunek 3: Czujnik BH1750

Podobnie jak w przypadku BMP180, sensor komunikuje się poprzez interfejs I^2C . Można go stosować do pomiaru natężenia światła zarówno w pomieszczeniach jak i w obszarze otwartym. Wejście **ADDR** sygnalizuje o wyborze adresu magistrali, w przypadku tego projektu nie musiało być używane.

3.4 Moduł WiFi ESP8266

Urządzenie to umożliwia bezprzewodową komunikację szeregową po WiFi do Internetu. Z całej rodziny opartych na chipie ESP8266, grupa wybrała model ESP-01. Działa on w standardzie WiFi 802.11 b/g/n na częstotliwości 2,4 GHz. Zasięg samego urządzenia to 300 metrów, a możliwość podłączenia go z lokalnym routerem posiadającym dostęp do Internetu sprawia, że posiadając zewnętrzny adres IP i zarezerwowany port może komunikować się z całym światem.



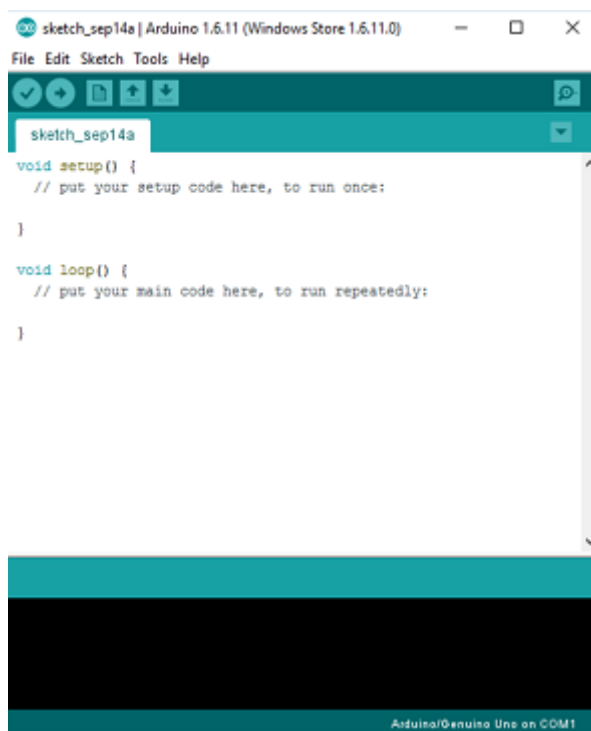
Rysunek 4: Moduł WiFi ESP8266

Model ten posiada dodatkowe wyprowadzenia GPIO (*general-purpose input/output*), które odpowiadają za komunikację z np. urządzeniami peryferyjnymi, więc nie przydają się w realizacji tego projektu. Wyjście VCC wymaga zasilania na poziomie 3,3V. **RX** i **TX** odpowiadają kolejno za odbiór i nadawanie danych. W dołączonym kodzie widać, że w połączeniu szeregowym na

Arduino za te funkcje odpowiedzialne są piny 2 i 3. W realizacji komunikacji modułu z resztą układu bardzo pomogła biblioteka *WiFiEsp.h*. Zapobiegała ona bezpośredniemu przesyłaniu komend AT, które ogólnie służą do komunikacji z modemami i są dosyć nieintuicyjne. Piny **CHPD** i **Reset** muszą być w stanie wysokim, więc również wędrują do zasilania 3,3V.

4 Programowanie

Arduino poprzez USB zostaje podłączone do komputera, gdzie za pomocą Arduino IDE można utworzyć kod programu komunikujący się z urządzeniem.



Rysunek 5: Środowisko programistyczne dla Arduino

Programy mogą być pisane w języku C/C++, jednak IDE znacznie ułatwia tworzenie kodu dzięki domyślnym bibliotekom, takim jak *Wiring*. Prostsze kody składają się przeważnie z dwóch części:

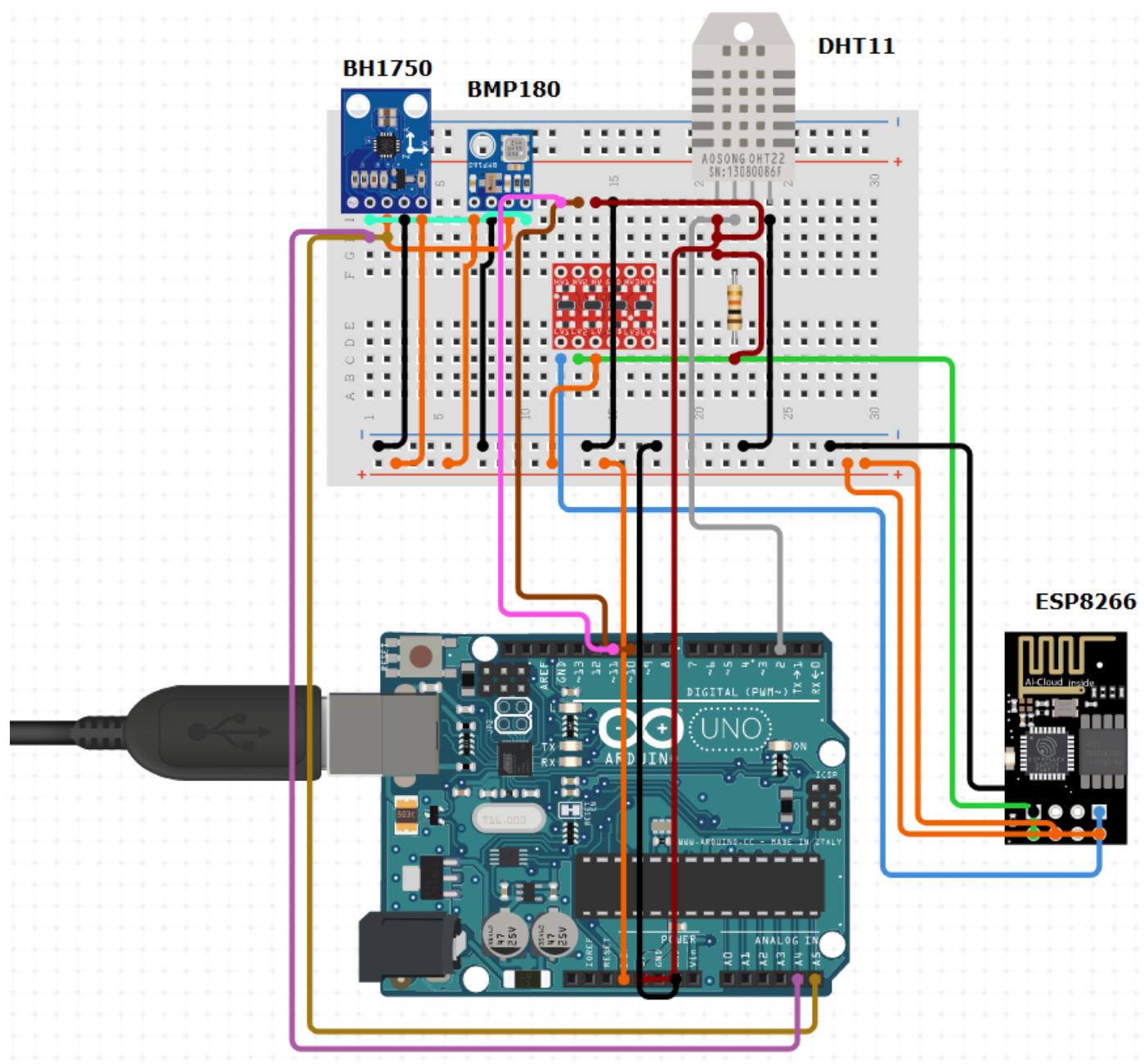
- `setup()` – funkcja wykonywana raz, na początku działania programu, wykorzystywana najczęściej do ładowania ustawień,
- `loop()` – funkcja wywoływana wielokrotnie, przez cały okres działania programu, czyli do czasu odłączenia zasilania od układu.

Do praktycznie każdego czujnika na stronie *GitHub* <https://github.com/adafruit> można znaleźć bibliotekę, która pomaga w sterowaniu danymi pobieranymi przez dany sensor. Znajduje się tam ponad 1000 repozytoriów do różnych urządzeń, a przykłady kodów dołączanych wraz z biblioteką powodują, że programowanie staje się łatwe i przyjemne.

Dane miejsca w kodzie podczas działania programu poprzez podłączoną płytkę można oglądać dzięki monitorowaniu portu szeregowego. Pokazuje on miejsca, które podczas realizacji kodu wypisują dane poprzez komendę: `Serial.print("przykładowa wiadomość")`. Dzięki temu programista może sprawdzić dane wysyłane np. poprzez czujnik DHT11, zanim będzie realizował dalszą część projektu.

Nie licząc języka C++, w programie został także użyty język znaczników HTML. Ministrona została nadawana poprzez moduł WiFi do klientów, chcących połączyć się i pobrać dane z serwera.

5 Schematy połączeń



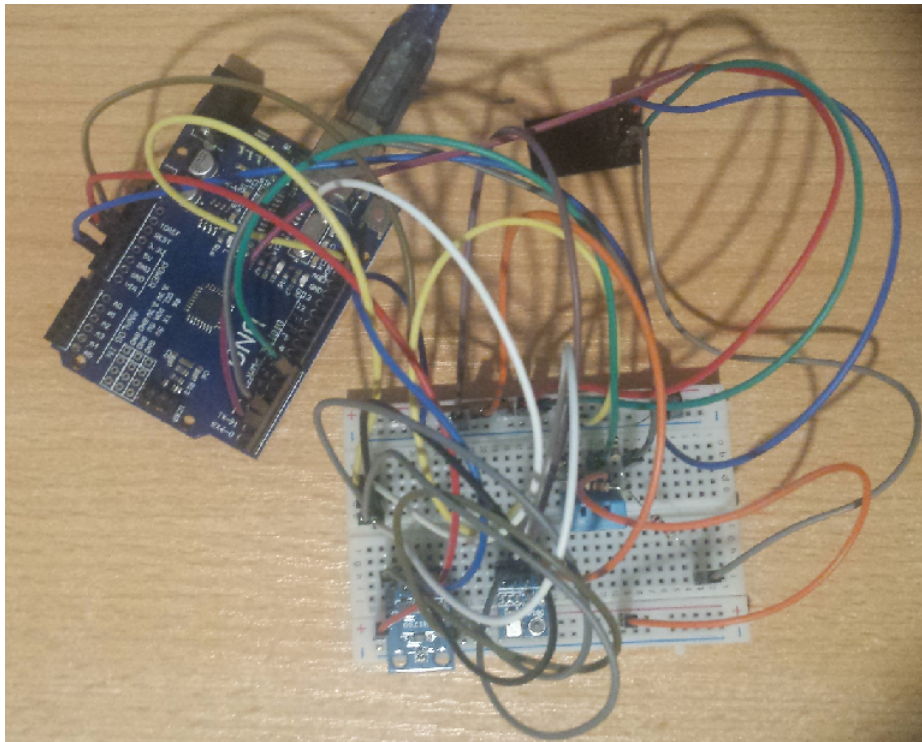
Rysunek 6: Schemat

6 Trudności w realizacji projektu

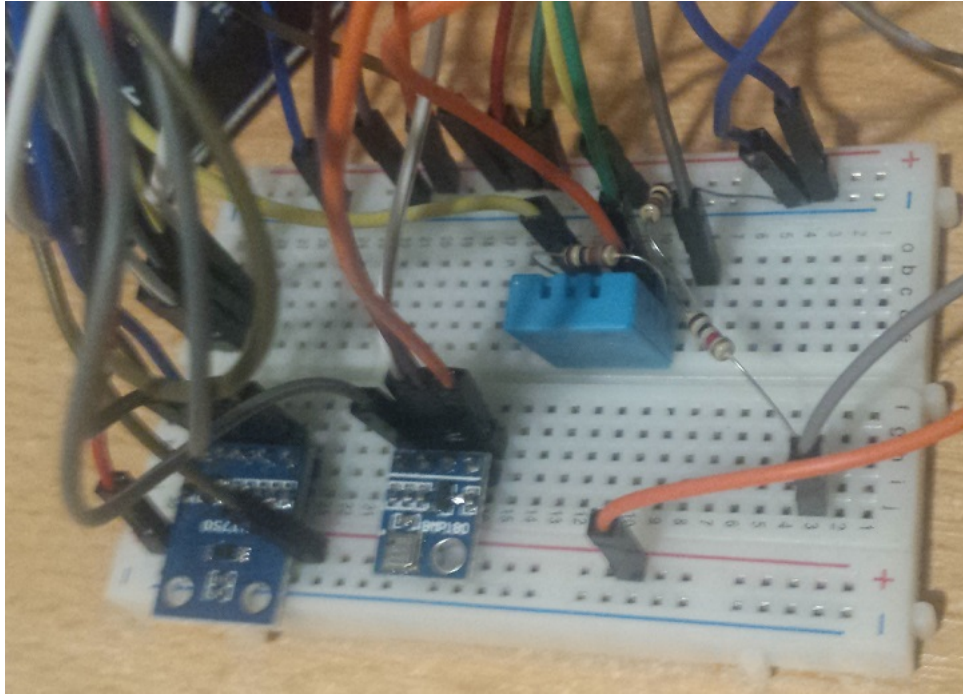
Grupa napotkała na trudności związane z fizycznym połączeniem niektórych części. Okazało się, że czujniki BMP180 i BH1750 posiadają piny niezłutowane z samymi płytkami, przez co grupa musiała złutować sensory za pomocą cyny w taki sposób, by pobór danych przebiegał bez zakłóceń. Była to dobra okazja do nabycia kolejnej ciekawej umiejętności.

Podpięcie wszystkich czujników do Arduino za pomocą kabli i płytki stykowej także nie należało do najłatwiejszych zadań, głównie przez mnogość kabli, co widać na zdjęciach w następnej sekcji. Bez odpowiednich narzędzi do segregacji kabli i schowania ich pod płytkę ciężko jest dostrzec, co za co odpowiada.

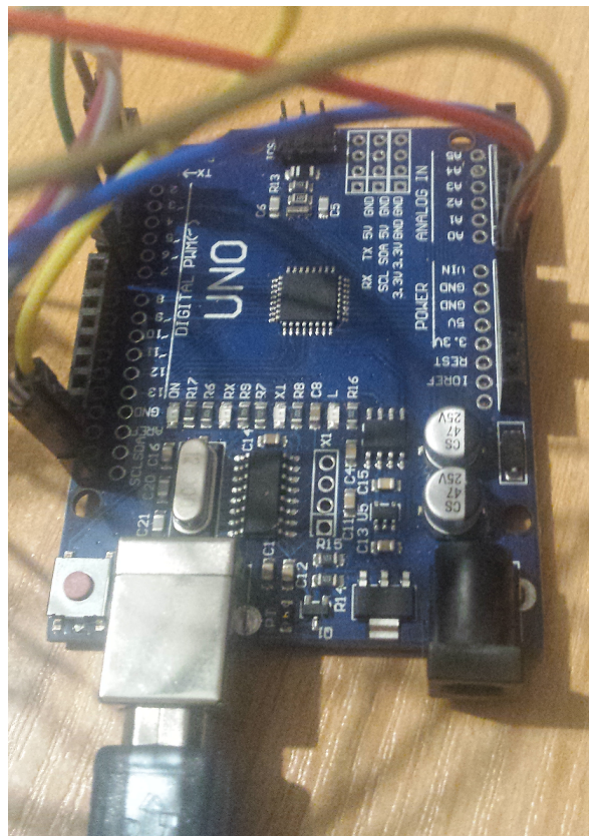
7 Zdjęcia systemu



Rysunek 7: Całość systemu



Rysunek 8: Moduły



Rysunek 9: Arduino

Witaj w moim domu :)

Temperatura: 24.00 °C
Wilgotnosc: 30.00 %
Cisnienie: 1014 hPa
Natezenie swiatla: 110.00 lx

LED: 0
Wejdz [tu](#) by wlaczyc (1)
Wejdz [tu](#) by wylaczyc (0)

Rysunek 10: Przykładowe pomiary na stronie

8 Wnioski

Projekt stacji pogodowej grupie udało się zrealizować bez większych przeszkód. Układ działa poprawnie i zgodnie z założeniami projektu. Po umieszczeniu systemu w odpowiedniej obudowie może on służyć jako autonomiczna stacja pogodowa.

Literatura

[1] <https://pl.wikipedia.org/wiki>

[2] <https://www.arduino.cc>