

Politechnika Śląska
Wydział Automatyki, Elektroniki i Informatyki
Raport końcowy z Systemów Mikroprocesorowych

PLANTIETM

Autorzy:
KAMIL CHOIŃSKI
OSKAR STABLA

6 stycznia 2020

Spis treści

1	Ogólny opis projektu	2
1.1	Cel i zakres projektu	2
1.2	Kosztorys	2
1.3	Opis podobnych rozwiązań	2
1.3.1	Jakis z neta	2
1.3.2	Jakiś z neta 2	2
2	Szczegółowy opis projektu	3
2.1	Rozwiązania techniczne - Przegląd modułów projektu	3
2.1.1	Czujnik wilgotności i temperatury powietrza: DHT11	3
2.1.2	Czujnik natężenia światła	3
2.1.3	Czujnik wilgotności gleby FC-28	3
2.1.4	Czujnik poziomu wody w zbiorniku	3
2.1.5	Wyświetlacz OLED	3
2.1.6	Przekaznik, Pompka wodna	4
2.1.7	Obudowa	6
2.2	Szczególne rozwiązania warstwy programowej	6
2.2.1	Płytką rozwojową UNO	6
2.2.2	ESP8266	8
2.2.3	Panel sterowania przez Internet: ESP8266	8
3	Realizacja projektu	8
3.1	Problematyka projektu	8
3.2	Sposób wykonania	8
3.3	Problemy napotkane podczas realizacji	9
3.3.1	Połączenie między UNO, a ESP8266	9
3.3.2	Projektowanie i drukowanie obudowy	10
4	Harmonogram i podział obowiązków	10
4.1	Podział obowiązków	10
4.2	Wstępny schemat blokowy	10
4.3	Harmonogram	11
4.3.1	4.11	11
4.3.2	25.11	11
4.3.3	16.12	11
4.3.4	20.01	11
4.4	Zgodność z harmonogramem	12
4.4.1	4.11	12
4.4.2	25.11	12
4.4.3	16.12	12
4.4.4	20.01	12
5	Podsumowanie	13
5.1	Osiągnięte cele	13
5.2	Pomysły na rozwój projektu	13

1 Ogólny opis projektu

1.1 Cel i zakres projektu

Nasz projekt ma na celu pomoc zabieganym ludziom, którzy nie mają czasu na zajmowanie się swoją ukochaną roślinką przez swój częsty brak pobytu w domu. Wystarczy dostęp do internetu, nic więcej.

Nasza koncepcja opiera się na systemie zdalnego zarządzania rośliną. Chcemy mierzyć parametry gleby i otoczenia takie jak wilgotność, nasłonecznienie. W zależności od odczytanych wartości przez płytkę rozwojową UNO połączoną z modułem ESP8266 będzie możliwe sterowanie pompką wody, lampą. Mamy zamiar połączyć projekt z IT, dlatego panel sterowania będzie umieszczony na stronie internetowej.

1.2 Kosztorys

Większość komponentów projektu została sprowadzona z Chin, ze względu na ich niską cenę. Niestety czas oczekiwania na nie wydłużył się i spowodował nieoczekiwane opóźnienia w harmonogramie.

Item	Cena
Płytkę rozwojową UNO	10 PLN
Moduł ESP8266	18 PLN
Czujnik wilgotności powietrza	5 PLN
Czujnik wilgotności gleby	5 PLN
Czujnik nasłonecznienia	5 PLN
Przetwornik	5 PLN
Żarówka	5 PLN
Pompa wodna	22 PLN
Kabelki, płytka uniwersalna, cyna, klej na gorąco	10 PLN
Suma	xxx PLN

Tablica 1: Zakupione produkty

1.3 Opis podobnych rozwiązań

1.3.1 Jakis z neta

x

1.3.2 Jakiś z neta 2

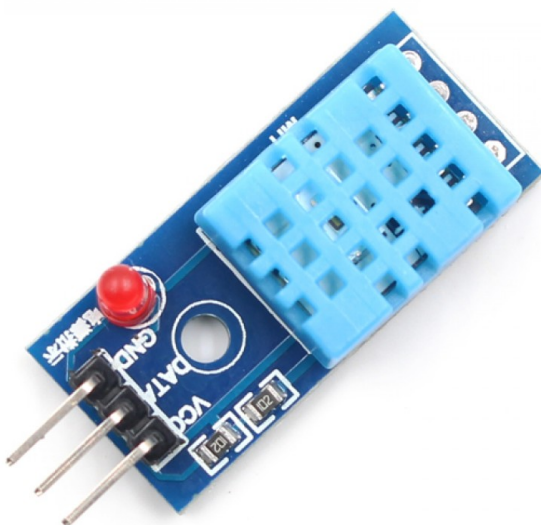
x

2 Szczegółowy opis projektu

2.1 Rozwiązania techniczne - Przegląd modułów projektu

2.1.1 Czujnik wilgotności i temperatury powietrza: DHT11

Pomiary wilgotności i temperatury powietrza są wykonywane z użyciem czujnika DHT11. Do implementacji w ArduinoIDE została wykorzystana biblioteka DHT-sensor-library, aby pomiary wykonywały się poprawnie.



Rysunek 1: Czujnik DHT11

2.1.2 Czujnik natężenia światła

Pomiary natężenia aktualnego oświetlenia w otoczeniu rośliny są wykonywane z użyciem czujnika natężenia światła.

2.1.3 Czujnik wilgotności gleby FC-28

Pomiary aktualnej wilgotności gleby w glebie naszej rośliny są wykonywane z użyciem czujnika wilgotności gleby FC-28.

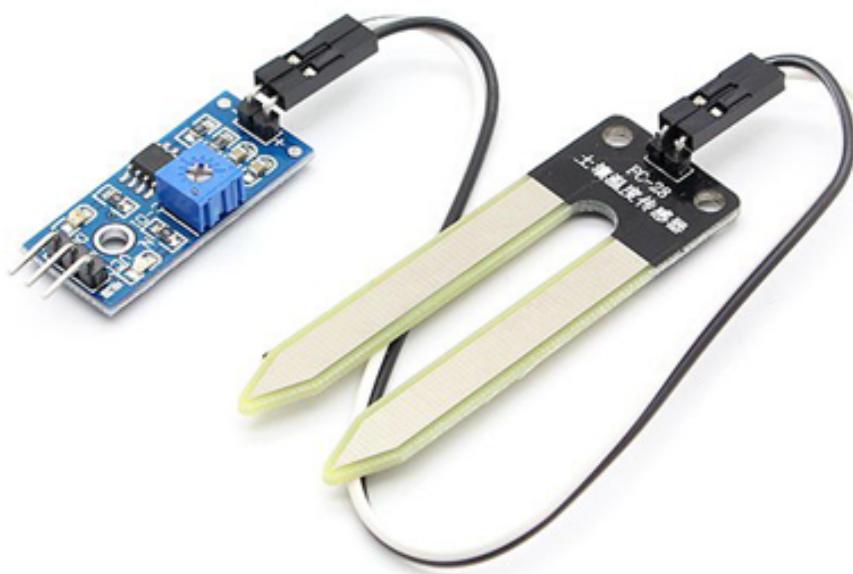
2.1.4 Czujnik poziomu wody w zbiorniku

2.1.5 Wyświetlacz OLED

Zewnętrzne wyświetlanie pomiarów wilgotności gleby i powietrza jest obsługiwane przez wyświetlacz OLED zamontowany na przodzie obudowy, aby była możliwość monitorowania jej stanu, gdy przebywamy w jej otoczeniu bez potrzeby połączenia internetowego. Do poprawnego wyświetlania danych na ekranie zostały wykorzystane biblioteki AdafruitGFX.h i AdafruitSSD1306.h.



Rysunek 2: Czujnik natężenia światła



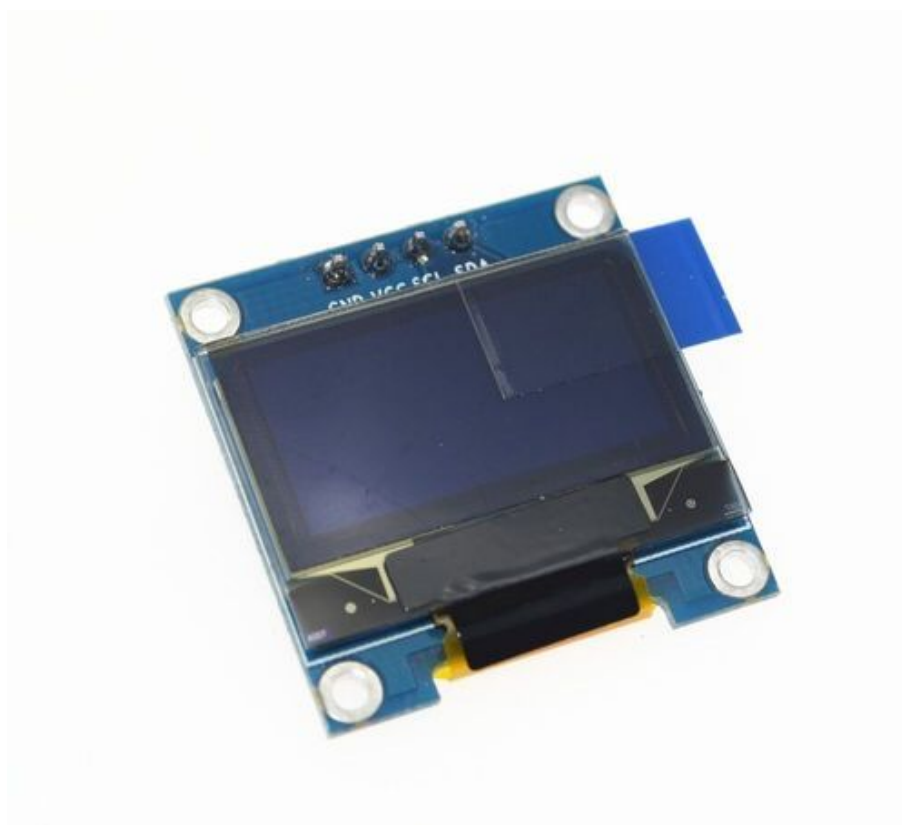
Rysunek 3: Czujnik FC-28

2.1.6 Przekaznik, Pompka wodna

Do układu została zamontowana samochodowa pompka do spyskiwaczy firmy TO-PRAN, gdyż było to tańsze rozwiązanie niż inne rodzaje pompek. Jej napięcie znamio-



Rysunek 4: Czujnik poziomu wody



Rysunek 5: 0.96 calowy wyświetlacz OLED

nowe wynosi 12 [V]. W projekcie do jej zasilania używamy baterii 9 [V]. Jej okresowe zasilanie jest realizowane za pomocą przekaźnika do którego z jednej strony wpięta jest



(a) Przekaznik [6]



(b) Pompa wodna [7]



(c) Bateria 9V [8]

Rysunek 6: Komponenty pompki

bateria i pompka, a z drugiej zasilanie 5 [V] z płytki rozwojowej UNO, GND i sygnał sterujący.

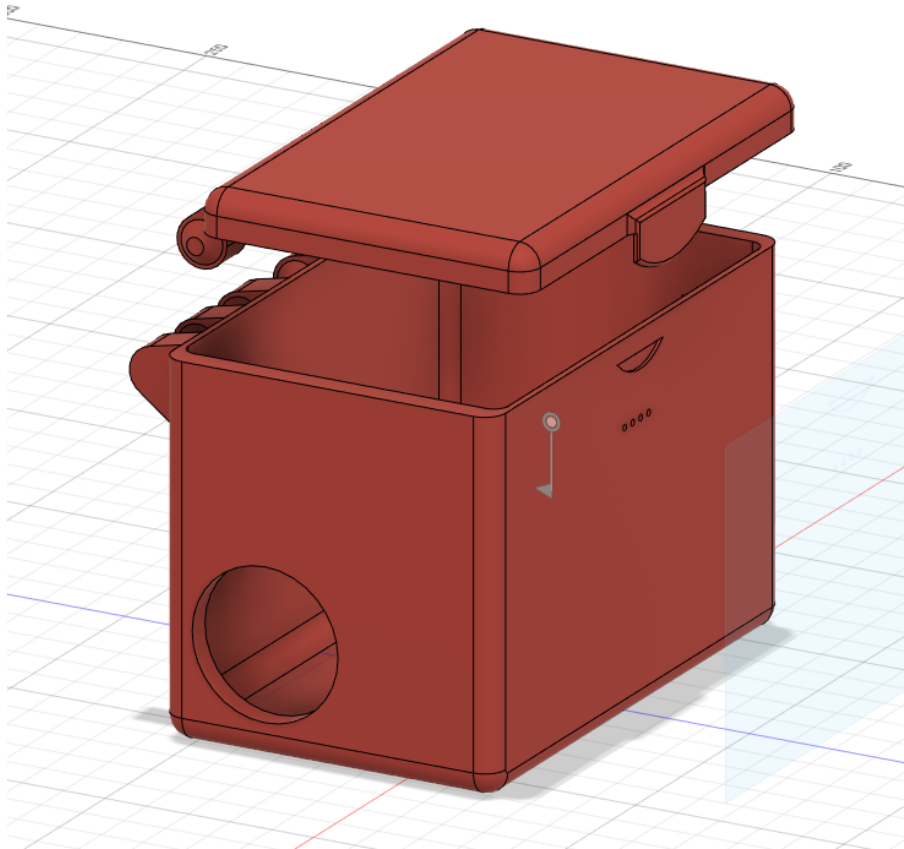
2.1.7 Obudowa

Obudowa na komponenty naszego projektu została zaprojektowana w programie Fusion360. Model ten został "pocięty" (przygotowany do druku) przy użyciu programu Cura i wydrukowany na drukarce 3D Ultimaker 2+. Został użyty materiał do drukowania o nazwie PLA (Polilaktyd) i dyszy wielkości 0.8mm. Czas drukowania pokrywki i pudełka wyniósł 12 godzin.

2.2 Szczególne rozwiązania warstwy programowej

2.2.1 Płytki rozwojowa UNO

Realizacja oczekiwania płytki rozwojowej UNO na sygnał sterujący z ESP8266 została wykonana poprzez ciągłe sprawdzanie czy mamy coś w jej buforze. Gdy wykryjemy że jest coś do odczytania zczytujemy ten znak. Dane które są odbierane będą zawarte w wskaźnikach początku "<" i końca ">" danej komendy, dzięki temu możemy wykryć kiedy zaczyna się nowa instrukcja i wyczyścić pozostałości jakie mogły pozostać w buforze, a także wykryć zakończenie danej instrukcji. Zapobiega to "gubieniu danych" i otrzymywaniu powielonych instrukcji.



Rysunek 7: Model obudowy (Fusion360)

Część kodu płytki rozwojowej UNO odbierająca dane

```
while (Serial.available())
{
    char serialChar = Serial.read();
    if (serialChar == '<')
    {
        serialMessage = "";
        serialMessage += serialChar;
    }
    else if (serialChar == '>')
    {
        stringComplete = true;
        serialMessage += serialChar;
    }
    else
    {
        if ((serialChar >= 48 && serialChar <= 122) || serialChar == ',')
            serialMessage += serialChar;
    }
}
if( stringComplete == true)
```



Rysunek 8: Płytką rozwojowa UNO

2.2.2 ESP8266

Połączenie płytki ESP8266 z Internetem jest realizowane z pomocą biblioteki ESP8266WiFi.h która ułatwia nam połączenie poprzez dodanie klasy WiFiClient, która przyjmując argumenty nazwy sieci Wi-Fi i hasła łączy się automatycznie z internetem. Gdy już uzyskamy połączenie z siecią Wi-Fi łączymy się z naszym serwerem na wcześniej zadeklarowanym IP i porcie za pomocą metody stworzonej z użyciem biblioteki WebSocketClient.h. Aby ułatwić testowanie po każdym nieudanym połączeniu restartujemy ESP, aby próbowało cały czas się połączyć.

Tak jak w płytce rozwojowej UNO, odbieranie i wysyłanie danych jest realizowane we wskaźnikach początku "<" i końca ">".

2.2.3 Panel sterowania przez Internet

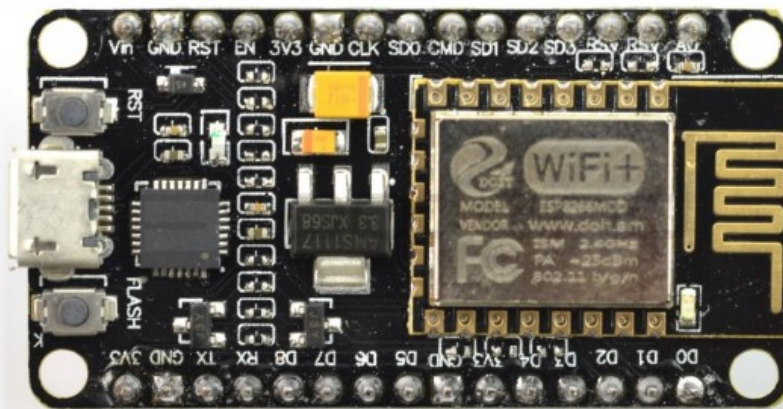
3 Realizacja projektu

3.1 Problematyka projektu

Przed rozpoczęciem prac nad projektem ..

3.2 Sposób wykonania

Postanowiliśmy połączyć płytkę rozwojową UNO obsługującą pompkę i monitorującą wartości naszych czujników z ESP8266, aby była możliwość sterowania poprzez Internet.



Rysunek 9: ESP8266

Płytką rozwojową UNO cały czas czeka na przychodzące dane które będą zawarte w wskaźnikach początku "<" i końca ">" danej komendy, aby wykonać określoną czynność lub odesłać pomiary wykonane przez czujnik wilgotności i temperatury do ESP8266, które jest połączone z UNO poprzez piny RX i TX. UNO odsyła swoje komendy również w wskaźnikach początku i końca, aby zapobiec "gubieniu" danych.

ESP8266 jest naszym tzw. "mostem" i w każdym momencie czeka na dane czy to z UNO czy ze strony internetowej i w zależności od kogo dane otrzymuje przerzuca je do danego odbiorcy. Program na ESP jest zrealizowany z użyciem bibliotek ESP8266WiFi, WebSocketClient, SoftwareSerial, które kolejno służą do: umożliwieniu połączenia się ESP do sieci WiFi zadeklarowanej w naszym kodzie, połączeniu się poprzez websocket jako klient do naszego serwera postawionego na laptopie, połączenia się poprzez piny RX i TX do płytki rozwojowej UNO.

3.3 Problemy napotkane podczas realizacji

3.3.1 Połączenie między UNO, a ESP8266

Próbowaliśmy różnych typów połączeń jak I2C, albo SPI jednak poprawność testowych danych jakie otrzymywaliśmy była nikła - albo dostawaliśmy tylko część danych albo były one w różny sposób powielone. Rozwiązaliśmy ten problem poprzez użycie

wskaźników początku "<" i końca ">" przy wysyłaniu danej komendy, aby wykonać określoną czynność lub odesłać pomiary.

3.3.2 Projektowanie i drukowanie obudowy

Dużym problemem była niewiedza związana z wielkością jaką powinna mieć obudowa, jednak zdecydowaliśmy że będzie ona większa niż będziemy potrzebować, aby umożliwić nam łatwe rozbudowanie projektu w przyszłości.



Rysunek 10: Nieudany druk pokrywki

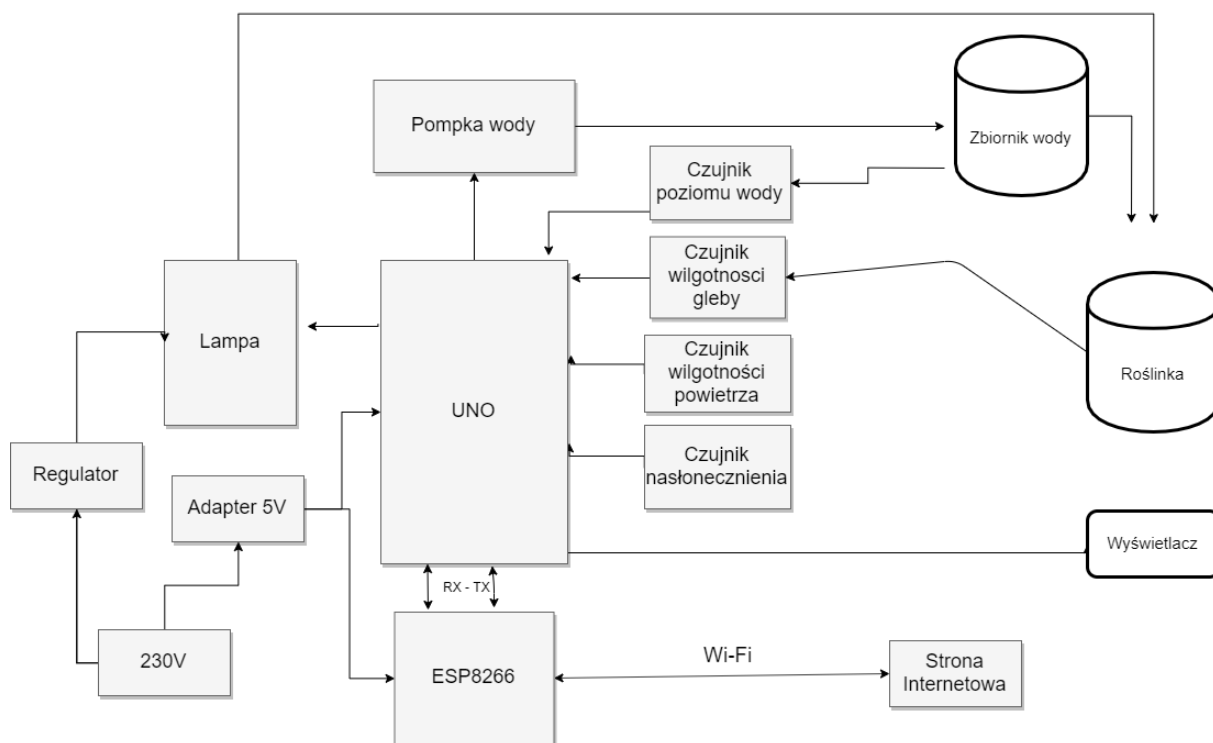
Przy początkach drukowania popełniliśmy błąd złej początkowej wysokości druku co spowodowało że model był drukowany w powietrzu a co za tym idzie druk się nie powiódł. Kalibracja drukarki 3D naprawiła nasz problem i druk został wykonany poprawnie

4 Harmonogram i podział obowiązków

4.1 Podział obowiązków

4.2 Wstępny schemat blokowy

Przed rozpoczęciem pracy nad projektem stworzyliśmy schemat, którego staraliśmy się trzymać jednak ze względu na pewne komplikacje w trakcie realizacji pewne aspekty zostały zmienione co zostało przedstawione na finalnym schemacie blokowym.



Rysunek 11: Wstępny schemat blokowy

4.3 Harmonogram

4.3.1 4.11

Przeanalizowanie schematów płytki rozwojowej UNO, modułu komunikacyjnego ESP8266, czujnika wilgotności gleby, czujnika wilgotności powietrza, czujnika nasłonecznienia oraz rozwiązanie techniczne doświetlania rośliny. Stworzenie schematu elektrycznego gotowego projektu. Testowanie poprawności działania posiadanych czujników w warunkach domowych. Implementacja komunikacji z modułem ESP8266. Dopasowywanie czasów działania. Dokupienie brakujących komponentów gotowego projektu.

4.3.2 25.11

Realizacja połączeń elektrycznych na płytce stykowej i ostateczne testowanie poprawności działania. Przeniesienie projektu na płytkę uniwersalną. Przygotowanie ewentualnej obudowy i realizacja montażu systemu doświetlania. Stworzenie prezentacji na zajęcia

4.3.3 16.12

Wstępna prezentacja gotowego projektu i ocena błędów.

4.3.4 20.01

Ostateczna prezentacja z poprawką błędów.

4.4 Zgodność z harmonogramem

4.4.1 4.11

Rozpoczęliśmy prace nad naszym projektem. Postanowiliśmy skorzystać z aplikacji internetowej Trello do utworzenia tabel, które pozwoliły by nam na lepsze zarządzanie podziałem prac nad projektem.

Przeanalizowaliśmy schematy płytki rozwojowej UNO, modułu komunikacyjnego ESP8266, czujnika wilgotności gleby, czujnika wilgotności powietrza oraz czujnika nasłonecznienia. Przeanalizowaliśmy działanie każdego z czujników poprzez wykonanie przykładowych kodów na płytce rozwojowej UNO. Dowiedzieliśmy się jakich komponentów projektu nam brakuje i zamówiliśmy je w wybranych sklepach. Stworzyliśmy wstępny schemat blokowy projektu, aby ukazać działanie poszczególnych elementów zawartych w naszym projekcie. Na ten moment postanowiliśmy nie tworzyć schematu elektrycznego, gdyż nie wiedzieliśmy jak dokładnie zrealizujemy poszczególne połączenia.

Ze względu na obszerność testowania i planowanie realizacji planu działania na nadchodzące tygodnie nie zajęliśmy się komunikacją z modułem ESP8266, ani problemem związanym z doświetlaniem naszej rośliny.

4.4.2 25.11

Do tego dnia zajmowaliśmy się komunikacją między płytką rozwojową UNO, ESP8266, a stroną internetową. Napotkaliśmy wiele problemów co nieplanowanie wydłużyło nam pracę na tym etapie projektu. Udało nam się uzyskać stabilne połączenie między naszymi urządzeniami. Zrealizowaliśmy połączenia UNO, ESP8266, a także podłączyliśmy do układu czujnik wilgotności i temperatury powietrza i na płytce stykowej po czym przetestowaliśmy poprawność działania.

Przeniesienie projektu na płytkę uniwersalną okazało się niepraktyczne na tym etapie projektu, gdyż nie próbowaliśmy jeszcze połączyć ze sobą wszystkich modułów projektu i finalny kod do wgrania na UNO i ESP8266 nie był gotowy. Z powodu niewiedzy jak dokładnie będzie wyglądało nasze urządzenie na płytce uniwersalnej nie przygotowaliśmy planowanej obudowy. Nie zrealizowaliśmy również systemu doświetlania.

4.4.3 16.12

Podłączyliśmy ze sobą wszystkie czujniki na płytce uniwersalnej i testowaliśmy poprawność działania. Zajęliśmy się projektowaniem obudowy w programie Fusion360, abyśmy mogli ją wydrukować na drukarce 3D i włożyć do obudowy komponenty naszego projektu. Wydrukowaliśmy obudowę i wsadziliśmy tam wszystkie rzeczy

4.4.4 20.01

Ostateczna prezentacja z poprawką błędów.

5 Podsumowanie

5.1 Osiągnięte cele

5.2 Pomysły na rozwój projektu

Literatura

- [1] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The L^AT_EX Companion*. Addison-Wesley, Reading, Massachusetts, 1993.
- [2] Ivan Grokhotkov *ESP8266 Arduino Core's documentation*. 2017
- [3] Christian Klippel, Peter Andersson, Peter Lerup *ESP8266 core for Arduino*. 2017
- [4] guy *w3schools.com*. 1999-2019
- [5] Albert Einstein. *Zur Elektrodynamik bewegter Körper*. (German) [*On the electrodynamics of moving bodies*]. Annalen der Physik, 322(10):891–921, 1905.
- [6] Relay Module , opencircuit.shop/Product/Relay-Module-1-relay.-5V
- [7] Pompa spryskiwacza TOPRAN , www.autoczescionline24.pl/topran-2723484.html
- [8] Bateria alkaliczna 9V VARTA, www.elfadistelec.pl/pl/alkaliczne-bateria-alkaliczna-9v-6lr61-varta-industrial-9v/p/16901614
- [9] Knuth: Computers and Typesetting,
<http://www-cs-faculty.stanford.edu/~uno/abcde.html>