



STEROWANA ZDALNIE UPRAWA HYDROPONICZNA

Projekt zaliczeniowy – Elektronika II

Jan Madejek, Piotr Dzioba, Jan Michalski
Mechatronika, WIM, II stopień, sem. 1, 2021/2022

13 czerwca 2022

Spis treści

1 Projekt systemu	2
1.1 Wstęp	2
1.2 Założenia	2
1.3 Kluczowe zadania	4
1.4 Inne pomysły	5
2 Budowa prototypu	6
2.1 Komunikacja – zarys idei, opis	6
2.1.1 Oprogramowanie	8
2.1.2 Sterowanie zdalne	8
2.1.3 Sterownik	10
2.2 Wybór czujników i kontrolera	11
2.2.1 Obliczenia poboru prądu	13
2.3 Konstrukcja	13
2.3.1 Obudowa	16
2.4 Kosztorys	17
3 Podsumowanie	19
3.1 Demonstracja możliwości	20
3.2 Trudności	22
3.3 Dalsze perspektywy	24
4 Bibliografia	26
A Rysunki wykonawcze i złożeniowe, schematy	27

Rozdział 1

Projekt systemu

1.1 Wstęp

Hodowla roślin w układzie hydroponicznym różni się w wielu aspektach od tradycyjnych rozwiązań. W systemie krąży roztwór specjalnej odżywki, w której zanurzone są korzenie. W ten sposób roślina może pobrać niezbędne substancje mineralne, a także rozpuszczony w wodzie tlen. Zamknięty obwód, w którym płynie roztwór, zasila zatem całą uprawę. Nie stosuje się ziemi.

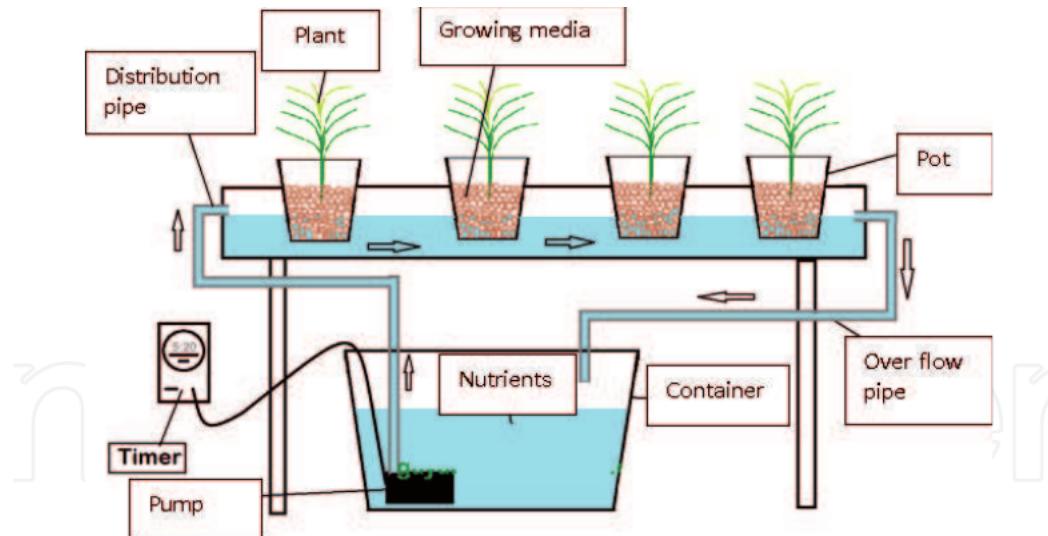
System hydroponiczny jest względnie łatwy do zautomatyzowania [1]. Można przyjąć, że roztwór jest homogeniczny, a zatem wystarczający jest jeden zestaw czujników, mierzących parametry wody w sposób ciągły. Dzięki temu, łatwo jest wyregulować odpowiedni poziom pH, natlenienia, kontrolować temperaturę wody. W przemysłowych rozwiązaniach, stosuje się także regulowanie poziomu CO₂. Warunki uprawy mogą zostać dobrane do ścisłych wymagań konkretnego gatunku, co przyspiesza wzrost i maksymalizuje wydajność. Sterowanie całym systemem można skoncentrować, a także łączyć wiele upraw i zarządzać nimi zdalnie. Automatyzacja jest tematem licznych prac badawczych [3][5].

1.2 Założenia

Projekt zakładał zbudowanie działającego prototypu systemu hydroponicznego, przeznaczonego do użytku domowego. Za przykładowe gatunki do uprawy, przyjęto przede wszystkim zioła, takie jak kolendra, bazylia czy mięta, oraz sałaty. Zaplanowano możliwość zdalnego sterowania parametrami pracy układu, a początkowo także implementację trybu automatycznego. Kompletny układ przemyślano tak, aby był w stanie zmieścić się do dostępnego w handlu detalicznym pojemnika.

Systemy hydroponiczne występują w wielu wersjach o tej samej metodzie działania, ale

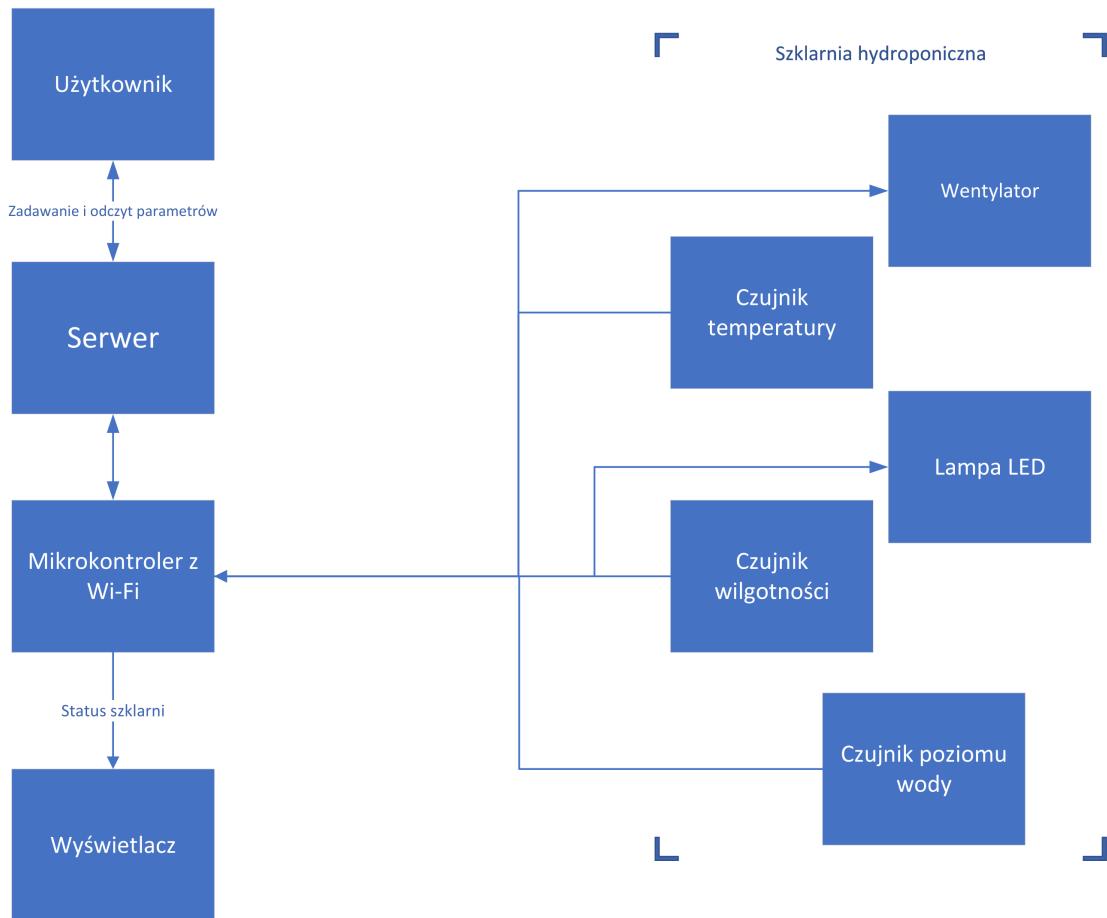
różniących się sposobem przepływu roztworu z pożywką. Wybrano układ NFT (ang. *Nutrient Film Technique*, pol. *Cienkowarstwowe kultury przepływowe*), który polega na użyciu kanałów z otworami, z umieszczonymi weń roślinami [Rys. 1.1]. Ciecz w tym systemie jest w ciągłej cyrkulacji, co zapewnia jej dobre zmieszanie, ułatwiające kontrolę parametrów roztworu, a także pomaga w natlenieniu. Ciągły ruch utrudnia także rozwój bakterii i grzybów. Korzenie mają relatywnie swobodny dostęp do powietrza (w stosunku do innych metod), ponieważ woda zajmuje niewielką część przekroju kanału. Jest to kluczowe założenie metody NFT.



Rysunek 1.1: Schemat układu NFT. [2]

Zdecydowano, że system w podstawowej konfiguracji, powinien posiadać możliwości:

- Pomiaru temperatury i wilgotności powietrza
 - Wykrywania zbyt małej ilości wody w zbiorniku
 - Sterowania bezprzewodowego
 - Naświetlania roślin specjalną lampą, wraz ze sterowaniem natężeniem światła
 - Wentylowania komory



Rysunek 1.2: Schemat ideowy

1.3 Kluczowe zadania

System powinien do minimum ograniczać konieczność osobistego nadzoru. Najważniejsze funkcje, które powinien realizować, to:

- Odczytanie i przetworzenie pomiarów z czujników
- Wyświetlanie statusu szklarni na wyświetlaczu
- Obsługa zdalnego sterowania manualnego
- Obsługa trybu automatycznego – przetwarzanie danych z czujników, podejmowanie decyzji, wykonywanie zadań wg harmonogramu.
- Wykonanie wizualizacji.

1.4 Inne pomysły

Oprócz uprawy hydroponicznej, rozważano także inne urządzenia i sposoby komunikacji dwustronnej, spełniające techniczne wymagania projektu. Z konceptem uprawy konkurował przede wszystkim projekt urządzenia alarmowego (ze względu na liczne włamania do komórek lokatorskich w ostatnim okresie w Poznaniu) oraz system bezpieczeństwa przeciwpożarowego do drukarki 3D.

Urządzenie	Parametry	Komunikacja
Ploter (laser/grawer)	Zadawanie ścieżki za pomocą aplikacji	??
Nadzór nad drukarką	temperatura, dym, czas, przekaźnik	serwer zbierający dane
Szklarnia-hydroponika	temperatura, wilgotność, PH, poziom wody, przewietrzanie	serwer zbierający dane
Projektor laserowy		aplikacja na telefon?
Głośnik BT		
Tracker do śledzenia skradzionych rzeczy	moduł GPS, moduł GSM, alarm do straszenia złodzieja?	GSM poprzez SMS
Alarm do skrytki lokatorskiej	Kontaktron, czujnik ruchu	LoRa, RFID
Rozwiązania IoT		
Google Cloud IoT	protokół MQTT, dwustronna komunikacja	
Google Firebase		
Thingspeak	klucze API	
Domoticz	system do domowej automatyki, po instalacji pluginów obsługa MQTT	
Thingsboard		
Home Assistant		
Platforma		
ESP-32 WROOM	niski koszt, WiFi i BT na pokładzie, programowanie przez Arduino IDE, piny do obsługi sterowania dotykowego	
Arduino+moduł WiFi		

Rysunek 1.3: Robocze notatki

Rozdział 2

Budowa prototypu

Do realizacji przekazywania danych między elementami systemu, konieczne było użycie wygodnego, najlepiej prostego w implementacji i popularnego, obsługiwanej przez narzędzia open-source protokołu. Należało także wybrać odpowiednie medium komunikacji. Rozpatrywano Bluetooth, Wi-Fi, a także LoRaWAN.

Od części konstrukcyjnej wymagano kompaktowości całego systemu, prostej budowy, niskiego kosztu części i elastyczności w kwestii modyfikacji i poprawek.

2.1 Komunikacja – zarys idei, opis

Po analizie dostępnych rozwiązań, wybrano protokół MQTT, który charakteryzuje się przejrzystością oraz niewielkim zużyciem przepustowości łączna. Jest popularną metodą do przesyłania telemetrii w rozwiązaniach Internetu Rzeczy. Oparty jest o wzorzec publikacja/subskrypcja.

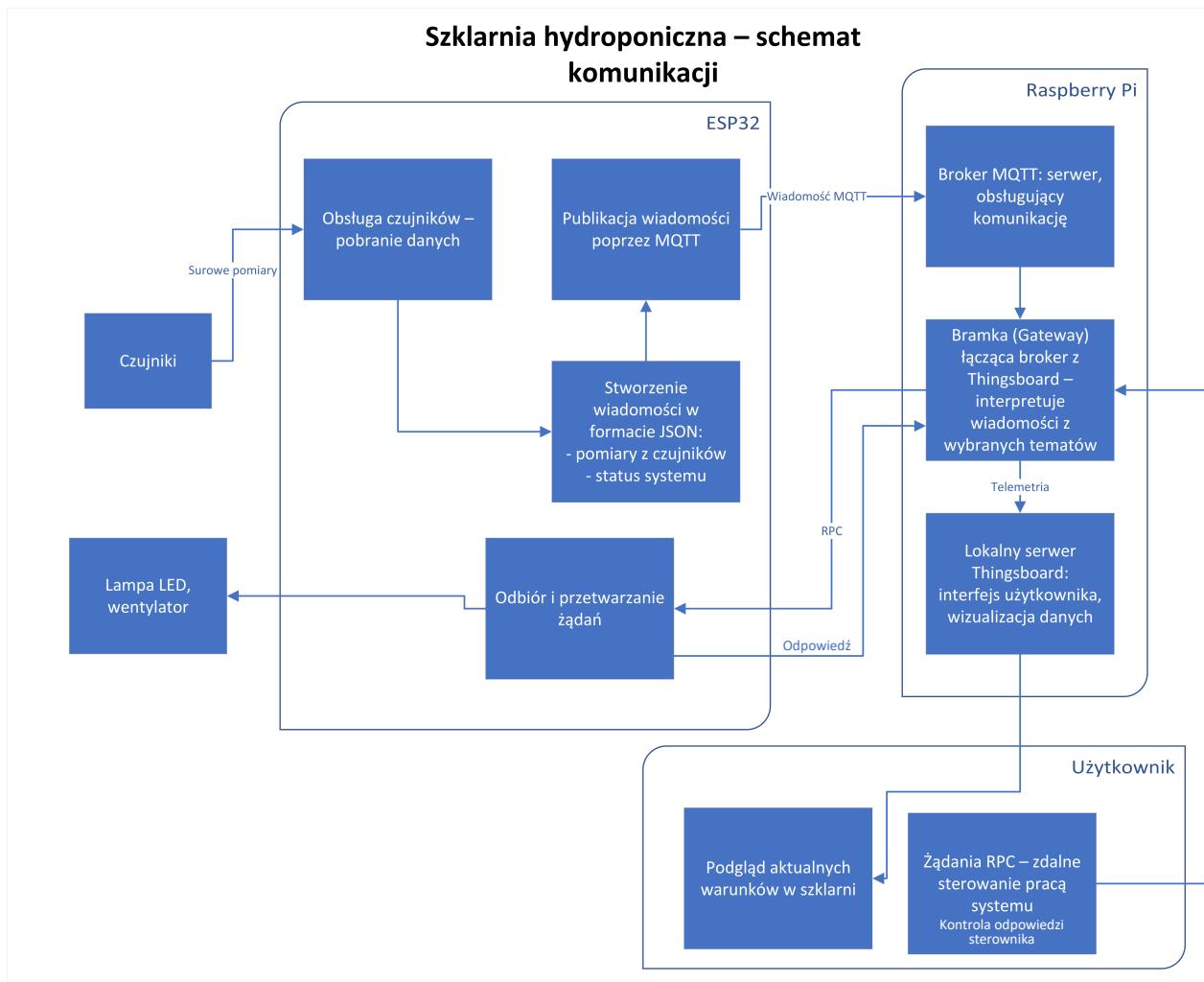
Początkowo, projekt planowano zbudować w oparciu o rozwiązania chmurowe, takie jak Google Cloud Platform, zapewniające bezpieczeństwo, mnogość możliwości przetwarzania danych, wspieranie innych rozwiązań Google i dostęp do szklarni z całego świata. Złożoność konfiguracji, w szczególności certyfikacji urządzeń, a także konieczność sprzęgnięcia ze sobą wielu API w celu uzyskania przepływu danych od urządzenia, przez bazę danych, aż do wizualizacji sprawiła, że wybrano rozwiązanie już wstępnie dostosowane. Jest nim platforma Thingsboard, przeznaczona dla IoT, wspierająca natywnie protokół MQTT. Oprócz niego, obsługiwane są także takie metody jak FTP, BLE, CAN i wiele innych.

Wymiana informacji odbywa się przy użyciu Wi-Fi, w obrębie lokalnej sieci. Nie jest wymagane połączenie z Internetem, system może działać całkowicie odizolowany od zewnętrznej sieci. Wi-Fi służy tutaj jako uniwersalne medium przekazywania danych, powszechnie obsługiwane.

wane przez w zasadzie każde urządzenie IoT.

Wspomniana platforma Thingsboard stanowi gotowy system do integracji i zarządzania urządzeniami IoT. Posiada także implementację przeznaczoną do instalacji na minikomputerze Raspberry Pi pod kontrolą systemu Raspbian, a także dla systemów Windows oraz Linux. Zastosowany model, Raspberry Pi 3B+ posiada wystarczającą wydajność, aby być w stanie zarządzać szeregiem procesów, niezbędnych do działania platformy. Należą do nich:

- Broker MQTT – serwer do publikacji wiadomości w tematach.
- Bramka MQTT Thingsboard (*gateway*) – przekazywanie i interpretacja informacji między brokerem, a platformą.
- Platforma Thingsboard – udostępnianie strony w lokalnej sieci, służącej do wizualizacji i sterowania pracą systemu.



Rysunek 2.1: Schemat blokowy komunikacji

Protokół MQTT, jak wspomniano, funkcjonuje we wzorcu publikacja/subskrypcja. Oznacza to, że klienci mogą publikować wiadomości w tzw. tematach, a także subskrybować dowolny

temat. Sterownik łączy się z brokerem Mosquitto, publikując następnie wiadomość ze statusem "połączono" w odpowiednim temacie. Platforma Thingsboard subskrybuje domyślnie ten temat, dzięki czemu jest w stanie rozpoznać i filtrować próby połączenia przez różnych klientów. Sterownik następnie, w temacie przeznaczonym na telemetrię, przesyła pomiary z czujników w ustalonym interwale. Jednocześnie, konieczne jest prowadzenie nasłuchu w temacie przeznaczonym dla komend wysyłanych poprzez platformę Thingsboard, co szerzej opisano w następnym podrozdziale [sekcja 2.1.2].

Wiadomości przesyłane są w formacie wymiany danych JSON, który zapewnia czytelność i jest łatwy do nauki.

2.1.1 Oprogramowanie

Tak jak wspomniano w Rozdziale 2.1, jako bazę do budowy systemu zastosowano platformę Thingsboard. Oprócz niej, wymagany do działania był także broker MQTT. Wybrano popularny program open-source Eclipse Mosquitto¹.

Cały kod do obsługi systemu napisano w języku C++, przy użyciu frameworku Arduino dla ESP32. Dzięki temu, możliwe stało się użycie bibliotek, a także funkcji i makr charakterystycznych dla Arduino, co przyspieszyło prace.

Ze względu na konieczność prowadzenia ciągłego nasłuchu żądań RPC [sekcja 2.1.2], niezbędne było napisanie programu w sposób nieblokujący, z uniknięciem opóźnień takich jak `delay()`. Z tego powodu użyto przerwań od timera. W pętli `loop()` umieszczono funkcję odpowiedzialną za nasłuch. W procedurze obsługi przerwań (ISR) znajduje się flaga `tim_flg`, następnie w głównej pętli zastosowano instrukcję warunkową, która jest wykonywana, gdy flaga ma wartość `true`. Ma to na celu uniknięcie wykonywania relatywnie długich operacji w procedurze ISR.

2.1.2 Sterowanie zdalne

Oprogramowanie Thingsboard umożliwia wprowadzenie tzw. żądań RPC (ang. *Remote Procedure Calls*). Są to komendy wysyłane na żądanie użytkownika, interpretowane przez sterownik. Przyciski umieszczone na tablicy kontrolnej tworzą wiadomości JSON z parametrami do nastawienia, a następnie publikowanie w temacie przeznaczonym na RPC.

Temat ten jest subskrybowany przez sterownik, a w razie przyjścia nowej wiadomości wywoływana jest funkcja `callback` [4]. Tam zlokalizowano procedurę odczytu nazwy tematu (każde kolejne żądanie ma inny identyfikator) oraz wybranej metody [Listing 2.2] i następnie rozkodowanie jej treści.

¹<https://mosquitto.org>

Po odebraniu wiadomości, sterownik odsyła potwierdzenie odbioru. Thingsboard przetwarza je, zapalając wówczas zieloną diodę sygnalizacyjną, widoczną na panelu sterowania.

Żądania do obsługi konkretnych urządzeń (zwane metodami) mają następującą strukturę:

Stan włącz/wyłącz

```
{  
    "method": "method_name",  
    "parameters":  
    {  
        "enabled": "state"  
    }  
}
```

Listing 2.1: Struktura żądania w JSON

gdzie `method_name` jest nazwą metody, a `state` to docelowy stan, przekazywany do urządzenia w formie `bool` o wartości `true` lub `false`.

Żądania do nastawy wypełnienia PWM do sterowania natężeniem oświetlenia mają inną strukturę, ze względu na brak możliwości edycji ustawień pokrętła regulacyjnego na tablicy kontrolnej. Publikowane są w innym, przeznaczonym tylko dla nich temacie, posiadają, tak jak poprzednio, unikalne nagłówki i zawierają tylko wartość w zakresie `< 0, 255 >` (8 bitów rozdzielczości).

```
char *request_id;  
char *ptr;  
char *methodName;  
ptr = strtok(topic, "/");  
  
uint8_t cnt = 0;  
while (ptr != NULL)  
{  
    ptr = strtok(NULL, "/");  
    cnt++;  
  
    if (cnt == 5) // w piątym+1 segmencie oddzielonym znakiem / znajduje  
    // się nazwa metody (wyjęta z nazwy tematu)  
    {  
        methodName = ptr;  
    }
```

```

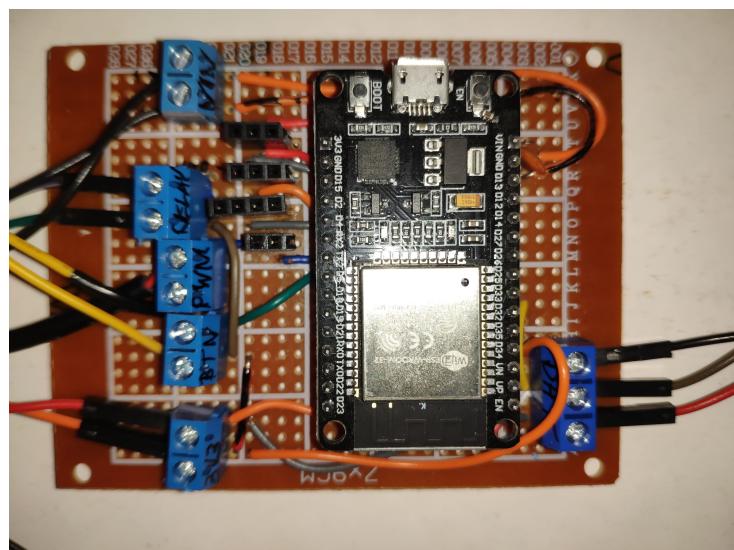
if (cnt >= 6)
{
    request_id = ptr; // analogicznie , kolejny segment zawiera
    // identyfikator RPC
    ptr = NULL;
}
}

```

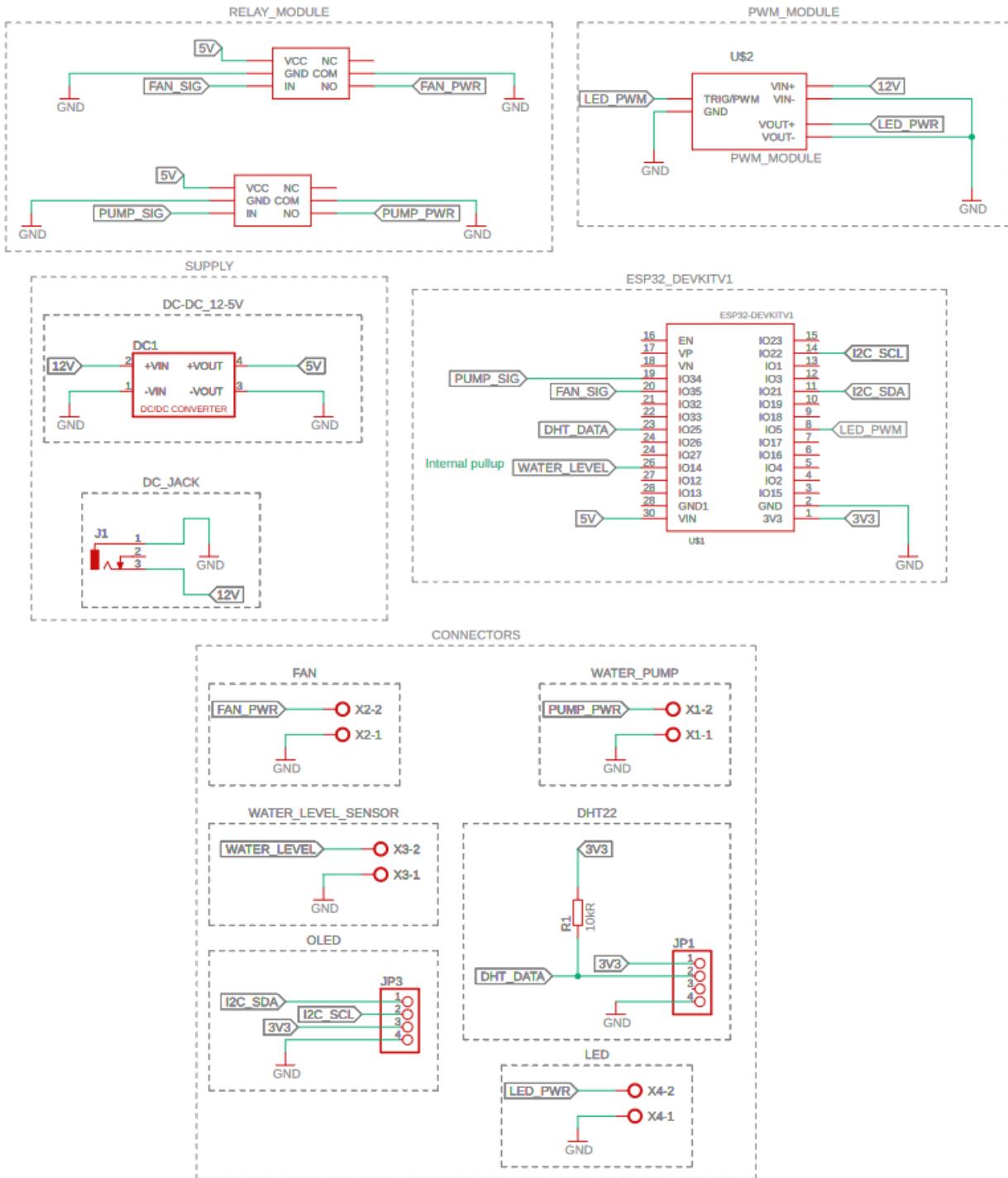
Listing 2.2: Fragment kodu funkcji callback

2.1.3 Sterownik

Ze względu na chęć rozwijania systemu poza ramami zajęć na Uczelni, wykorzystano płytkę prototypową do budowy części elektronicznej. Taka płytka zapewnia łatwość wprowadzania zmian i poprawek.



Rysunek 2.2: Płytki sterownika



Rysunek 2.3: Schemat elektryczny

2.2 Wybór czujników i kontrolera

W celu uproszczenia konstrukcji, starano się wybrać komponenty tak, aby charakteryzowały się napięciem zasilania równym 12 V. Ze względu na to, że płytka DevKit z układem ESP32 wymaga napięcia zasilania równego 5 V, użyto przetwornicy step-down.

Tabela 2.1 przedstawia końcową listę części, zastosowaną w projekcie. Na rys. 2.4 znaj-

duje się lista robocza, która służyła za tablicę do konfrontacji propozycji wyboru komponentów.

Element	Model	Cechy	Zalety
Czujnik temperatury powietrza	DHT22	Zakres: 40 do 80 °C, Rozdzielczość: 8-bit, Dokładność: ± 0,5 °C	Łatwe podłączenie (rodzaj interfejsu OneWire), wystarczająca rozdzielczość i dokładność pomiaru
Czujnik wilgotności powietrza	DHT22	Zakres: 0 do 100 %RH, Rozdzielczość: 8-bit, Dokładność: ± 2 %RH	j.w.
MCU	ESP32	Napięcie zasilania: 5V (stabilizator na 3,3V wbudowany w płytę) Taktowanie: maks. 240 MHz Pamięć flash: 4 MB 30 pinów GPIO	Wbudowany układ Wi-Fi, framework Arduino ułatwiający programowanie, obsługa przycisków dotykowych
Lampa do roślin	AKB Poland Grow LED	Zasilanie 12V, moc $14,4 \frac{W}{m}$, diody czerwone i niebieskie	Spektrum dostosowane do potrzeb roślin, dwustronna taśma na odwrocie, łatwe docięcie na pożądaną długość
Czujnik poziomu cieczy	ESCO GMF-21	Max. napięcie przełączane 200 VDC, max. prąd 0,7 A, mocowanie na pręt gwint. z nakrętką, sposób działania: kontaktron	Niewielkie rozmiary, niski koszt, pomiar binarny jest wystarczający
Przetwornica step-down	Oparta na LM2596	$V_{in} \in < 3; 35 > [V]$, $V_{out} \in < 1,5; 30 > [V]$, $I_{max} = 2[A]$, zabezpieczenie nadprądowe	Wystarczająca wydajność prądowa, niewielki rozmiar, wysoka sprawność
Moduł PWM	Oparty o N-Mosfet AOD4184A	Napięcie odbiornika $V_{out} \in < 5; 36 > [V]$, $I_{out} \leq 15[A]$	Niewielkie rozmiary, wystarczające parametry, niska cena

Tabela 2.1: Wybór najważniejszych komponentów

Szklarnia hydroponiczna			
Element	Model	Ilość	Linki
akwarium/pudełko/zbiornik	IKEA Samla	1 - https://allegro.pl/oferta/pojem	Dla uproszczenia proponuję, żeby wszystkie sprzęty zasilać 12V DC
pompka napowietrzająca do akwarium		1 - https://www.plantica.pl/akwan	Musimy też ogarnąć, jaki rodzaj szkłami chcemy - z obiegiem wody/bez, zamknięta/otwarta itd.
kamień/listwa napowietrzającej/coś podobnego		1 https://www.plantica.pl/akwarystyka-slodkowodna/napowietrzanie/kamienie-i-listwy/	
węze do pomp			- https://www.psautomatyka.com.pl/pl/p/Przewod-poliuretanowy-PU-niebieski/189
pompa wody?		1 https://allegro.pl/oferta/pompki	IP68 - super
allegro czujnik temp.	DHT22	1 - https://bottland.com.pl/czujniki-multifunkcyjne/9301-czujnik-temperatury-i-wilgotnosci-dht11-50c-590442237	
allegro czujnik wilgotności	DHT22		Jeden czujnik do temperatury i wilgotności
czujnik Ph?		1 https://kamami.pl/czujniki-ph/	Nie wiem czy nie zrezygnować z ph, dosyć drogie te czujniki
xkom lampa led		1 - https://new-electric.pl/2865-3_MU1z-SMGL8KLJ7Jlm924y6E1kaApNQEALw_wCB	
wentylator?	dowolny komputer	1 https://www.micross.com.pl/produkt/W%20205010v1s1_14066.html?gclid=CjwKCAjwxOCRBhA8EiwA0Xhi044	
przekaźnik	Moduł przekaźnik >=3		
allegro wyświetlacz		1 https://allegro.pl/oferta/357-wyswietlacz-oled-0-91-ssd1306-arduino-i2c-10481210541?snapshot=MjAyMi0wNl	
może przyciski jakieś? ułatwia też debugowanie			
allegro mikrokontroler	ESP32-devkit	1	
allegro moduł PWM	AOD4184A	1 https://sklep.avt.pl/modul-sterownik-pwm-400w-15a-36v-mosfet.html?gclid=CjwKCAjwopWSBhB6EiwAjxmqD	
allegro przetwornica 12-5 V	LM2596	1 https://allegro.pl/oferta/przetwornica-3a-dc-dc-step-down-lm2596-dla-arduino-9563532985	

Rysunek 2.4: Robocza lista części

2.2.1 Obliczenia poboru prądu

Zapotrzebowanie prądowe obliczono dla najmniej korzystnego przypadku, tzn. założono, że wentylator, lampa LED i pompka będą pracować jednocześnie na maksymalnej dostępnej wydajności, a układ ESP32 będzie połączony z Wi-Fi i wysyłał jednocześnie informacje.

Model	Pobór prądu [mA]
DHT22	średnio 0,2
ESP32	ok. 80
Wentylator	120
Lampa LED	2520 dla 2,1 m taśmy
Pompka	375
Wyświetlacz	max. 25

Tabela 2.2: Pobór prądu dla najgorszego przypadku

Zasilacz zatem powinien posiadać wydajność prądową $\geq 3,12[A]$, co po zaokrągleniu w górę, daje wymaganą moc zasilacza $\geq 40[W]$.

2.3 Konstrukcja

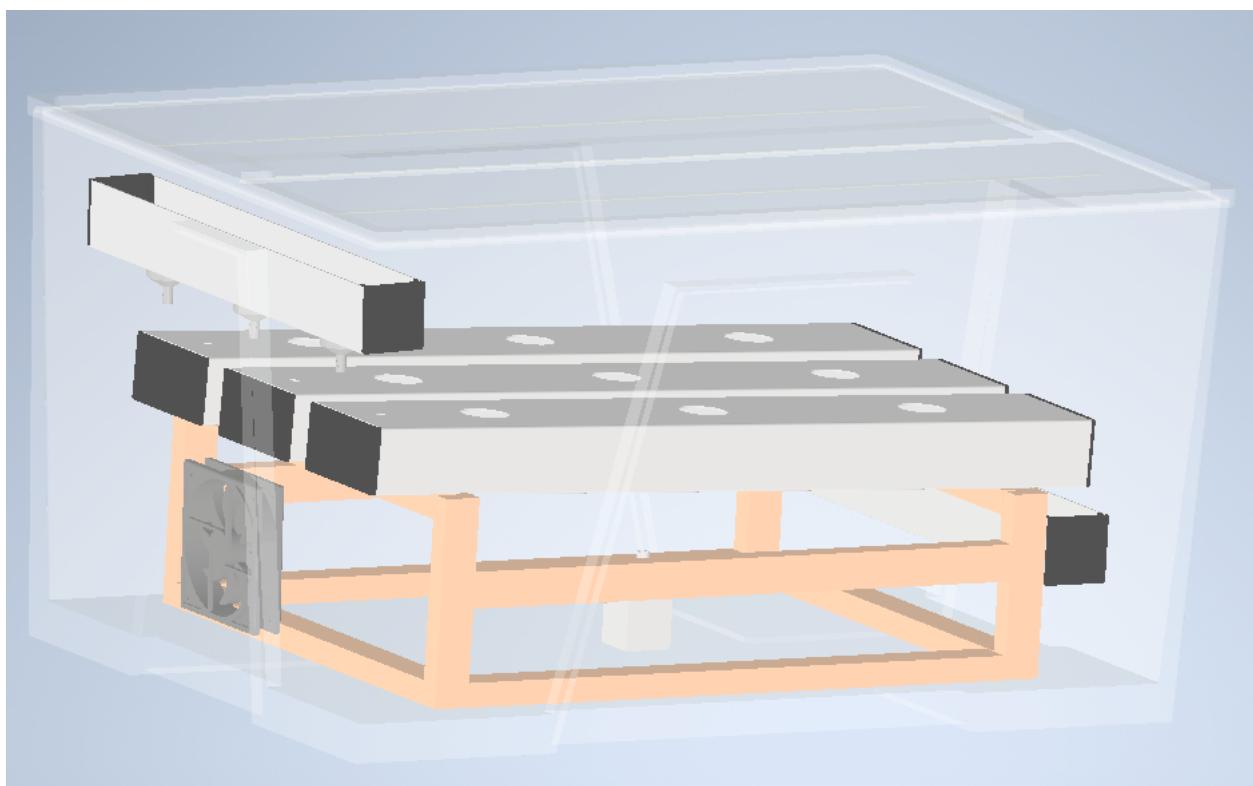
Całą przestrzeń szklarni hydroponicznej tworzy pojemnik z tworzywa sztucznego o pojemności 130 litrów wraz z pokrywką. Szklarnia wyposażona jest w stelaż wykonany z sosnowych kantówek o przekroju 20x30 mm, do którego przytwierdzone są kanały. W kanałach wykonano

otwory na sadzonki. Końce kanałów zostały zabezpieczone dekielkami oraz uszczelnione. Do wykonania modelu CAD wykorzystano oprogramowanie Autodesk Inventor.

Na potrzeby projektu wykonano model króćca, przeznaczonego do montażu w ścianie pojemnika i posiadającego gniazdo do przyłączenia węża.

W szklarni hydroponicznej zastosowano obieg wody przez kanały przy użyciu kolektora zbiorczego, pompy oraz pojemnika z króćcami, z którego woda spływa do kanałów. Cały układ jest połączony systemem węży gumowych. Do jednej ze ścian pojemnika przytwierdzono wentylator w celu regulacji przepływu powietrza [Rys. 2.7] oraz temperatury. Do spodniej strony pokrywki przyklejone zostały taśmy LED z regulacją natężenia oświetlenia ułatwiające przeprowadzenie procesu fotosyntezy. Szklarnia wyposażona jest w czujnik temperatury i wilgoci oraz czujnik poziomu wody w zbiorniku. Sterownik został zamknięty w obudowie w celu jego ochrony przed wpływem środowiska, w którym się znajduje.

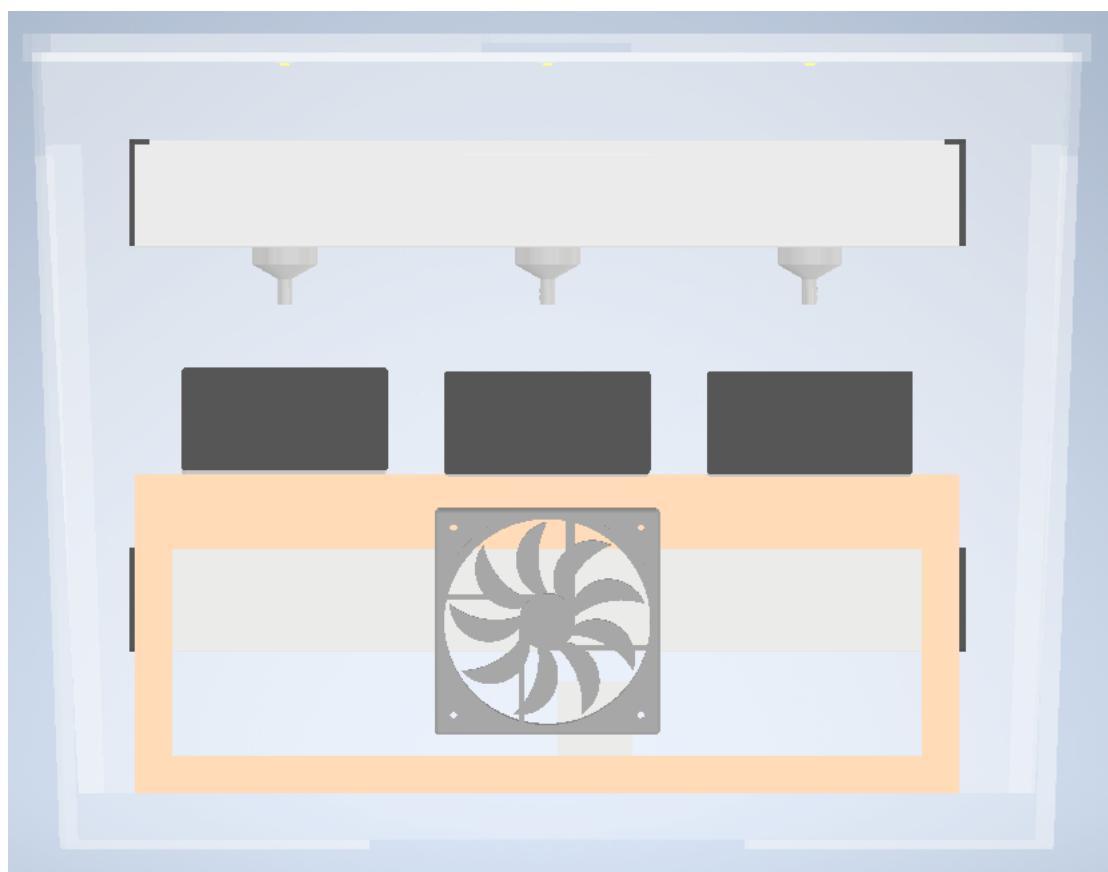
Na Rys. 2.6 widać niewielkie pochylenie kanałów. Jest to zabieg niezbędny w hydroponicznych systemach NFT, ze względu na konieczność pozbycia się zalegającej wody, niebędącej w ruchu. Zastosowano otwarte wyjścia kanałów uprawnych, co jeszcze bardziej poprawia przepływ. Użycie króćca z węzem mogłoby doprowadzić do powstania zatorów, a w konsekwencji do gnicienia korzeni.



Rysunek 2.5: Rzut 3D



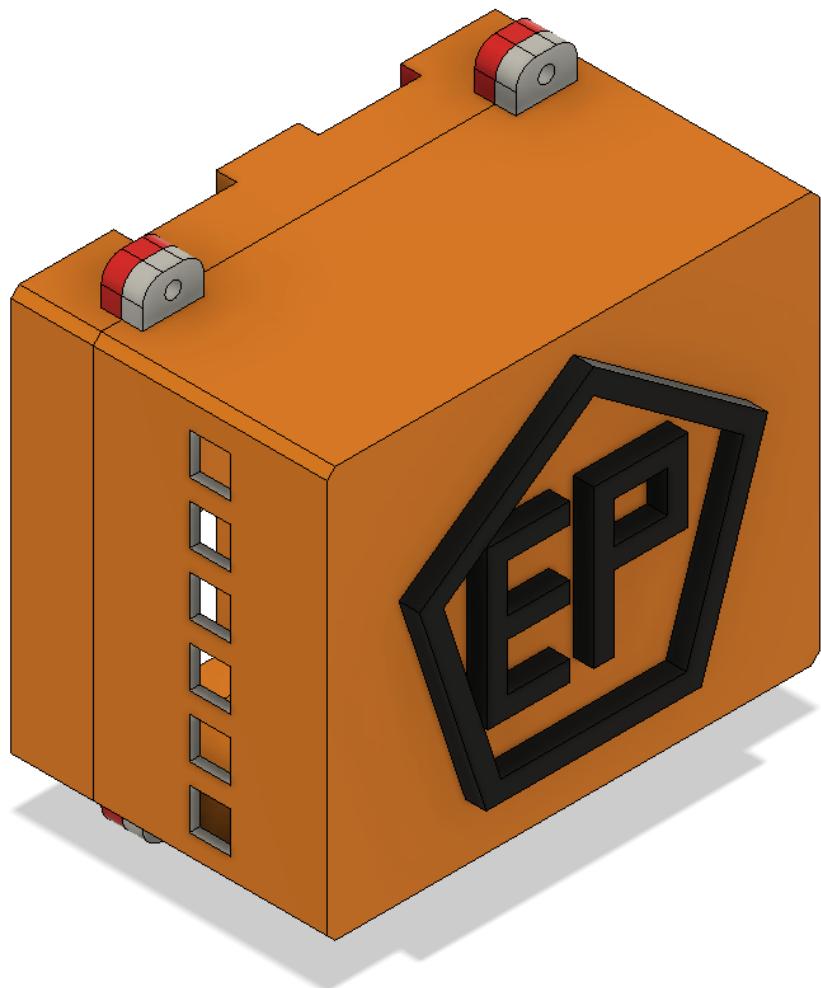
Rysunek 2.6: Rzut boczny



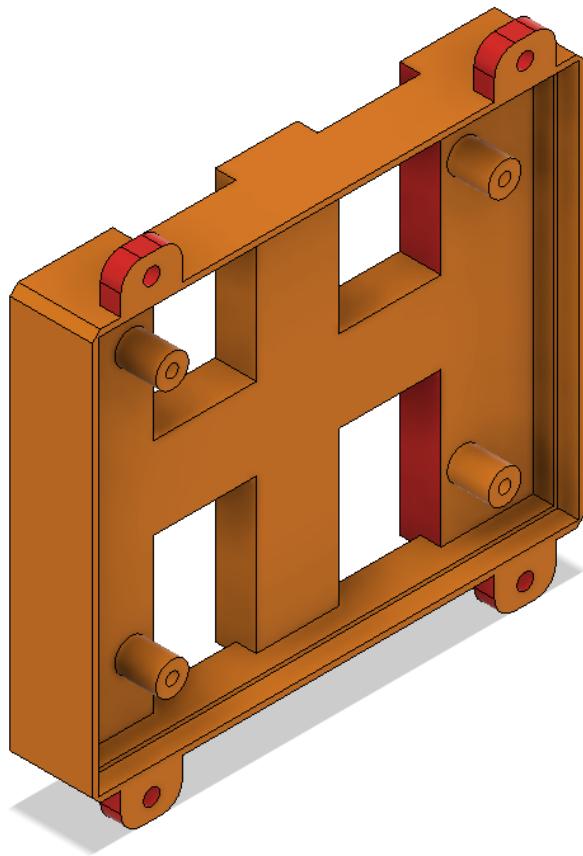
Rysunek 2.7: Rzut od tyłu

2.3.1 Obudowa

Obudowę sterownika [Rys. 2.8] przygotowano z myślą o wytworzeniu przy pomocy technik przyrostowych. Składa się z bazy [Rys. 2.8], do której przykręcana jest płytka prototypowa oraz pokrywy, która jest podstawą do montażu niektórych modułów.



Rysunek 2.8: Obudowa sterownika



Rysunek 2.9: Podstawa obudowy sterownika

Na ścianach bocznych znajdują się otwory na przewody, które służą także do wentylacji. Obudowę planuje się przymocować do ściany pojemnika przy użyciu taśmy montażowej dwustronnej.

2.4 Kosztorys

Sumarycznie, zakup części elektronicznych niezbędnych do wykonania sterownika, oraz elementów do budowy samej szklarni, wyniósł 416,88 zł. W cenę wliczono także zakup niektórych produktów, jak wkręty, które nie zostały wykorzystane w całości. Koszt natomiast nie obejmuje komputera Raspberry Pi, który należał już wcześniej do jednego z Autorów.

Element	Cena [zł]
Uniwersalna płytka PCB 7x9cm jednostronna	4,76
Zestaw zworek do płytki stykowej	13,16
Przetwornica 3A DC-DC step-down LM2596	7,16
Wyświetlacz OLED 0,91 SSD1306	15,71
Moduł MOSFET PWM 36V 400W 15A	5,71
Czujnik poziomu cieczy ZP5210	11,71
Czujnik temperatury i wilgotności DHT22	25,71
Pompka wody 240l/h 12V	29,71
Wężyk	7,40
Kanały wentylacyjne	57,00
Pojemnik IKEA SAMLA 130l + pokrywka	99,99
Kantówki drewniane	23,70
Wkręty 3,5x30	3,31
Wkręty 3,5x40	4,07
Taśma montażowa	20,90
Zaślepki do kanałów	36,12
Króćce	10,00
Przewody	20,00
Klej epoksydowy	15,99

Tabela 2.3: Kosztorys

Rozdział 3

Podsumowanie

Zaprojektowanie i wykonanie prototypu przyniosło wiele cennej wiedzy z różnych dziedzin. Projekt wymagał integracji elektroniki, programowania, mechaniki, a zatem dobrze wpisywał się w ideę mechatronicznego podejścia do rozwiązywania problemów. Korzystano z modułów o dość ogólnym zastosowaniu, które po połączeniu w jeden system, stanowiły synergiczną całość o przeznaczeniu specjalistycznym.

Projekt był także dobra okazją do przetestowania nowych narzędzi do współpracy: niżejszą dokumentację napisano w procesorze L^AT_EX, co było dla Autorów pierwszą większą pracą w tym środowisku. Pierwszy raz także skorzystano z framework'u Arduino dla ESP32, który okazał się prosty i skuteczny w zastosowaniu. Współpraca pod względem programistycznym prowadzona była w systemie kontroli wersji GitHub, projekt posiada własne repozytorium: https://github.com/mich-j/ESP32_hydroponics, do którego odwiedzenia Autorzy serdecznie zapraszają. W celu komunikacji między autorami wykorzystano Google Drive, który pozwolił na płynny dostęp do zawartości projektu oraz możliwość jego aktualizacji w sposób widoczny dla wszystkich autorów.

Do zrobienia	
Wykonanie modelu	DONE
Zbudowanie prototypu	DONE
Zbudowanie płytka ze sterownikiem	DONE
Model płytka	DONE
Oprogramowanie komunikacji serwer->ESP	DONE
Zakup napowietrzacza	
Przygotowanie dokumentacji	DONE
Schemat komunikacji	DONE
Montaż prototypu	DONE
zamodelowanie złączek do węzy	DONE
zamodelowanie kolektora do kanałów	DONE
model zaślepek do kanałów	DONE
wykonanie złączek	
<hr/>	
Zakupy	
Zakup węzy do pomp	DONE
Zakup kantówek	DONE
Zakup rezerwuaru	
Zakup wkrętów do drewna	DONE
Zakup taśmy montażowej	DONE

Rysunek 3.1: Plik do wspólnej kontroli postępów prac

Sposób komunikacji okazał się właściwy dla potrzeb projektu. Zapewnia dużą elastyczność pod względem rozmieszczenia elementów systemu (ograniczenie to jedynie zasięg sieci Wi-Fi). Opóźnienie reakcji sterownika jest niewielkie, a platforma Thingsboard pozwala na szeroką konfigurację tablicy *dashboard* oraz tworzenie estetycznych wizualizacji.

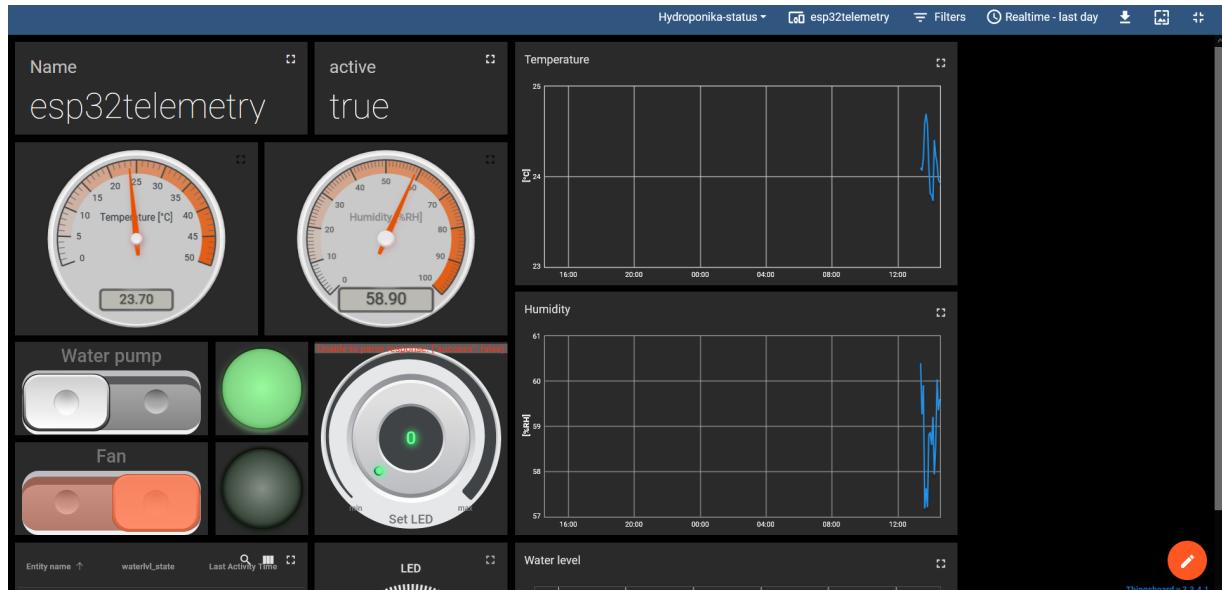
3.1 Demonstracja możliwości

Projekt realizuje obecnie funkcje:

- Weryfikacja, czy sterownik (ESP32) jest połączony z systemem
- Zdalny podgląd parametrów z wizualizacją w formie wykresów, historia do 30 dni wstecz:
 - Temperatura i wilgotność powietrza
 - Poziom cieczy (binarnie)
- Zdalne nastawianie i kontrola odpowiedzi sterownika:
 - Jasność lampy LED
 - Włączanie/wyłączanie pompy wody oraz wentylatora

Rysunki 3.2 i 3.3 prezentują *dashboard*. Widoczne są wykresy temperatury i wilgotności

powietrza oraz wskaźniki zegarowe. Obok ulokowano suwaki i pokrętło do sterowania zdalnego. Zielone diody sygnalizują uruchomienie urządzenia, natomiast w przypadku lampy LED o regulowanej jasności, aktualny stan wskazuje półkolisty sygnalizator. Jeżeli sterownik nie odbierze żądania, diody i wskaźnik jasności nie zaświecą się, a status "active" w górnej części będzie wskazywał "false".



Rysunek 3.2: Dashboard. Podgląd pomiarów z ostatniego dnia



Rysunek 3.3: Dashboard. Podgląd pomiarów z ostatnich 30 dni

Poniżej odsyłacz do krótkiego filmu, prezentującego działanie elementów systemu:

<https://www.youtube.com/watch?v=qL8lwmbou20>

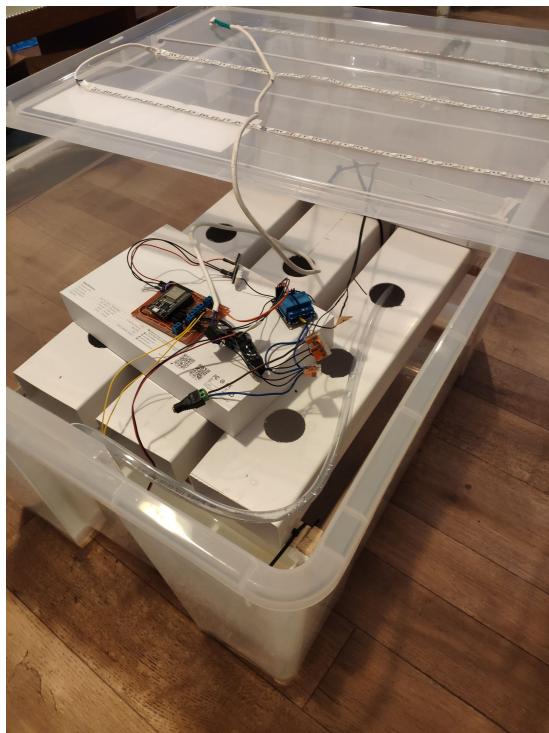
3.2 Trudności

Nie udało się zrealizować wszystkich założeń projektowych. Obsługę wyświetlacza OLED oprogramowano w szczegółowej formie (aczkolwiek został on uruchomiony i przetestowany), ze względu na skupienie na innych aspektach projektu. Oprogramowanie systemu zajęło więcej czasu, niż początkowo założono – po wstępny skonfigurowaniu Google Cloud Platform porzucono tą metodę, co zmusiło do poświecenia czasu na zapoznanie się z inną. Wykorzystanie Raspberry Pi jako serwera wymagało zapoznania się z obsługą poprzez terminal z użyciem SSH, nauczeniem się podstawowych komend powłoki *bash*, a także struktury folderów w systemie Raspbian.



Rysunek 3.4: Wydrukowana, niezamontowana obudowa

Bardzo wymagająca i czasochłonna okazała się część konstrukcyjna. Doszło do niespodziewanych zdarzeń które sprawiły, że system nie działa tak, jak zostało to zaplanowane. W trakcie testów doszło do awarii pompy wody, jednak wcześniej wykonano kilka prób, które pozwoliły na uzyskanie przepływu przez system kanałów.



Rysunek 3.5: System w trakcie testów

W początkowej koncepcji szklarnia miała zostać pozbawiona stelażu. Kanały miały być mocowane w otworach wyciętych w pojemniku. Stwierdzono jednak, że wycięcie odpowiednich otworów przy użyciu dostępnych narzędzi może być problematyczne dlatego zrezygnowano z tego pomysłu na rzecz wykonania stelażu i przytwierdzenia do niego kanałów. Pozwoliło to na zastosowanie krotszych kanałów. Dzięki temu rozwiązaniu udało się umieścić cały układ obiegu wody wewnątrz pojemnika. Dzięki temu cała szklarnia jest zamknięta w jednym pojemniku. Ułatwia to transport i zwiększa mobilność szklarni.

```
# wyświetlenie logów
sudo tail -n20 -f /var/log/mosquitto/mosquitto.log

# kontrola statusu mosquitto podczas pracy
sudo service mosquitto status

# edycja pliku z konfiguracją brokera
sudo nano /etc/mosquitto/conf.d/connect.conf

# wylistowanie katalogów
sudo ls

# wyświetlenie aktywnych procesów
ps -ef

# sprawdzenie statusu portu 1884
netstat -anp | grep 1884

# zakończenie procesu
sudo kill -kill <pid>

# edycja pliku z konfiguracją bramki MQTT-Thingsboard
sudo nano /etc/thingsboard-gateway/config/mqtt.json
||
```

Rysunek 3.6: Zanotowane podstawowe komendy *bash*

3.3 Dalsze perspektywy

Planowane jest dalsze rozwijanie projektu. Doświadczenia z budowy prototypowego systemu posłużą jako baza do konstrukcji nowej, ulepszonej wersji, bądź głębskiej modyfikacji obecnej. Przewiduje się nadrobienie niezrealizowanych założeń i wprowadzenie kolejnych funkcji:

- Możliwość uruchomienia trybu automatycznego.
- Wprowadzenie możliwości hodowli roślin o większej wysokości, przez montaż dodatkowego stelażu.
- Stworzenie modułu do wzrostu sadzonek z nasion.
- Wykonanie stelaża z profili montażowych, co zwiększy jego dokładność wymiarową w stosunku do projektu i stabilność (ale zwiększy też masę). Profile umożliwiają budowę większego systemu.
- Wprowadzenie czujnika pH wody, co rozszerzy spektrum kontrolowanych parametrów.
- Uzupełnienie pomiarów o temperaturę wody.

Konieczne będzie także poprawienie szczelności układu i mocowań kanałów. Część elementów zostało umocowanych przy użyciu taśmy montażowej, co jednak utrudnia zmiany i modyfikacje. Korzystniejsze byłyby połączenia rozłączne, np. śrubowe.

Rozdział 4

Bibliografia

- [1] *Hydroponics*. eng. IntechOpen, 2012. ISBN: 953-51-0386-5.
- [2] Alejandro Luna-Maldonado. “Automation and Robotics Used in Hydroponic System”. In: Dec. 2019.
- [3] Manav Mehra et al. “IoT based hydroponics system using Deep Neural Networks”. eng. In: *Computers and electronics in agriculture* 155 (2018). Place: OXFORD Publisher: Elsevier B.V, pp. 473–486. ISSN: 0168-1699.
- [4] Nick O’Leary. *Pub/Sub client*. May 2020. URL: <https://pubsubclient.knolleary.net/>.
- [5] Kebin XU et al. “Construction of an Automatic Nutrient Solution Management System for Hydroponics—Adjustment of the K+-Concentration and Volume of Water”. eng. In: *Analytical sciences* 35.5 (2019). Place: TOKYO Publisher: The Japan Society for Analytical Chemistry, pp. 595–598. ISSN: 0910-6340.

Dodatek A

Rysunki wykonawcze i złożeniowe, schematy

1. Rysunek złożeniowy
 - 1.1 Stelaż
 - 1.2 Szklarnia hydroponiczna ze stelażem
 - 1.3 Króciec
2. Rysunki wykonawcze:
 - 2.1 Elementu 1 króćca
 - 2.2 Elementu 2 króćca
 - 2.3 Koryta do uprawy
 - 2.4 Rezerwuar
 - 2.5 Koryto kolektora
3. Rysunki pomocnicze
 - 3.1 Płytką prototypowa
 - 3.2 Koncept montażu rynienek
4. Schematy
 - 4.1 Schemat elektryczny + plik źródłowy Autodesk Eagle
 - 4.2 Schemat komunikacji

4.3 Schemat ideowy